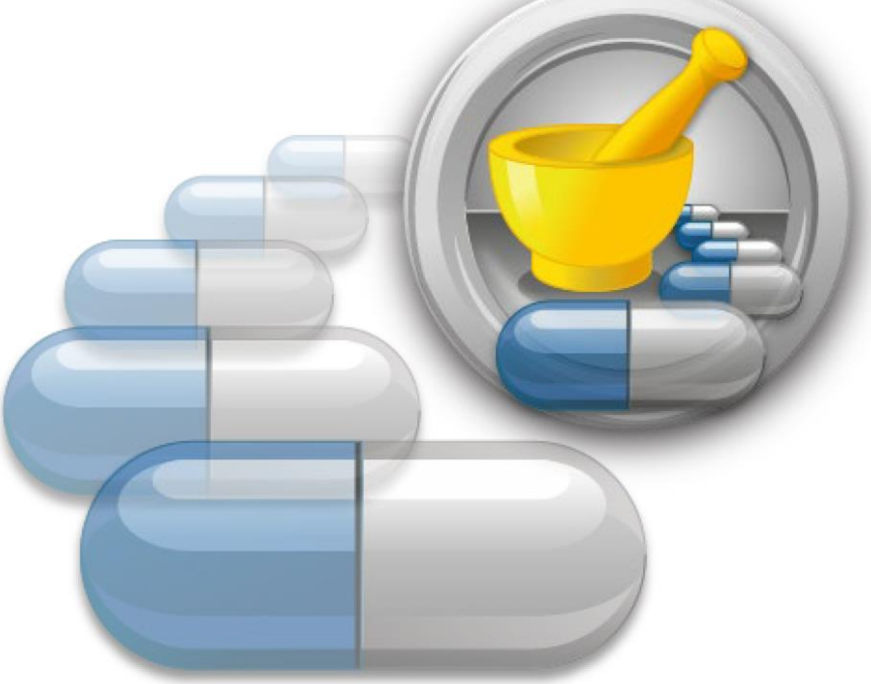
FT PharmaSuite®



RELEASE NOTES

RELEASE 10.02.00

PUBLICATION PSRN-RN010C-EN-E-SEPTEMBER-2021
Supersedes publication PSRN-RN010B-EN-E

Rockwell Automation

Contact Rockwell See contact information provided in your maintenance contract.

Copyright Notice

Trademark Notices

Other Trademarks

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

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

Contents

Figures

Introduction

This document provides background information in terms of "Information and Tips" (page 5), a "List of Known Issues" (page 37), and "Upgrade-related Topics" (page 41). All are important regarding the use of PharmaSuite 10.02.00

In addition, section "Release-specific Enhancements and Changes" (page 3) provides an overview of new key features since the previous release PharmaSuite 10.01.00.

Please see appendices for Technical Release Notes (page 49) and Open Source License Agreements (page 53).

For a list of definitions, please refer to the Glossary [A1] (page 45).

Chapter 2

Release-specific Enhancements and Changes

PharmaSuite 10.02.00 introduces functional improvements in the areas of:

- Production Execution Client Enhancements
    - Notification and Alarm Panel
      Displays notifications triggered by an ETO creation or an ETO escalation or by the new **Send User Notification** phase. An alarm panel allows the recognition of alarming notifications from a greater distance.
      A notification displays a recipe-specific text and a timestamp or a timer counting down to 0:00.
    - Visualization of ETO escalations
      Escalation capability parameters in Recipe and Workflow Designer allow to define if an escalation shall be displayed on the new notification panel and if it is an alarming notification.
    - Escalation management for a started ETO
      Finish due and finish overdue times can be now defined as escalation capability parameter for started operations. This can trigger an escalation to remind the operator to complete a started operation within a specified period of time after its creation.
- New Production Execution Viewer (PEV)
  The Production Execution Viewer provides a complete overview of an order or workflow that is currently in process or has already been processed. Thus, it can support various roles in various scenarios:
    - an operator who works on several stations simultaneously, to determine on which station an upcoming task needs to be completed next.
    - during shift change, to ensure a smooth hand-over by providing the new shift with an overview of the current order.
    - a support person, to access processed data of previous operations or unit procedures in order to understand and quickly resolve issues during production.
- Recipe and Workflow Designer Enhancements
    - Embedded Export/Import functionality for recipes, workflows, and building blocks.

    - Allow recipe transfer between sites: Allowing a recipe import with missing master data. Removed references are documented and visible as warning messages.
- Exception Review Enhancements
    - Exceptions with the **None** risk class without status handling
    - Change of closed exceptions with additional access privilege only
    - PRC Explorer Tree: Initially opened on UP level. Allow to collapse/expand including all levels by the keyboard shortcut.
    - Get DCS Alarms phase with a configurable representation of the converted and unconverted alarms within the related exception.
- Automation Integration Server
    - Load monitoring via JMX
    - Improved OPC Group handling

Limitations

The scope of PharmaSuite 10.02.00 is further limited as described in the following paragraphs.

- Backup and Recovery
  Backup and recovery (SR1075.6) is done using 3rd-party tools, therefore not scope of the testing activities for PharmaSuite 10.02.00.
- Supported Platforms
  If a requirement states that a specific software or hardware configuration shall be “supported” rather than “qualified”, this configuration will be informally tested. Therefore, supported configurations are not qualified in terms of any risk analysis or formal tests.
  The supported and qualified platforms are denoted in the "PharmaSuite Supported Platforms Guide" [A22] (page 45).
- System Configuration
  Changing system configuration and / or extending the system, which is typically done by system integrators, is not tested for PharmaSuite 10.02.00. Testing for PharmaSuite 10.02.00 is generally limited to a default configuration, unless otherwise mentioned.
- Internationalization (I18N)
  I18N aspects are explicitly excluded from testing because of negligible risk.

Chapter 3

Information and Tips

This section provides information and tips on the behavior of PharmaSuite under specific conditions.

General Considerations

Maintenance of User Master Data

Updated for PharmaSuite 9.1.

User master data of PharmaSuite is maintained with PS Administration. Typically, on project level a third-party security provider like the LDAP Login Module is used.

However, FactoryTalk ProductionCentre still requires that user-specific properties and user group assignments are maintained in PS Administration, even if integrated with third-party applications. This is specifically required for a user's first name and last name. The information is mandatory, since it is used across all PharmaSuite clients whenever the name of a user has to be recorded (e.g., logged-in user, electronic signature).

For more information, please refer to "Implementation Guide PS Administration" [A14] (page 45).

Precision of Calculation

Updated for PharmaSuite 8.3

If a measured value with its given scale (number of significant digits) is multiplied by a factor, the resulting scale depends on the scale of the original value and the magnitude of the factor (smaller than or greater than 1).

Examples:

- The expected result of a weight of 35.5 g multiplied by 10 is **355 g** (not: 355.0 g).
- The expected result of a weight of 7 kg multiplied by 0.5 is **3.5 kg** (not: 4 kg).

In order to achieve the expected result, PharmaSuite applies an algorithm to correct the result's scale as expected:

- scaleTarget = ceil(scaleSource - log10(multiplierFactor))

This algorithm also has to be considered when tolerances are defined in a recipe (e.g. with process parameters of the **Get process value** phase).

- The following tolerance definition **must not be used**, because it results in different and unexpected scales for upper/lower limits (example tolerance of +/-1%):
  Upper limit = Value * (1 + 0.01)
  Lower limit = Value * (1 – 0.01)
  - Example: Value = 58 ml
    Upper limit = 59 ml, Lower limit = 57,4 ml
- The following tolerance definition results in a **correct and expected representation** of the limits (example tolerance of +/- 1%):
  Upper limit = Value + (Value * 0.01)
  Lower limit = Value - (Value * 0.01)
  - Example: Value = 58 ml
    Upper limit = 58.58 ml, Lower limit = 57,42 ml

Related to percentage calculations in general, the scale of a percentage calculation result will be at least 2 digits higher than the resulting scale of the multiplication itself due to the implicit division by 100. Example: 5 * 5% = 0.25, 100 * 1% = 1.00.
For more information, please refer to "Technical Description - Number Handling" [A10] (page 45) and "Technical Description - Quantity Calculations" [A11] (page 45).
The "Technical Description - Number Handling" describes the handling of numbers with respect to unit of measures (UoM), conversions, (quantity) calculations, etc. in PharmaSuite.
The "Technical Description - Quantity Calculations" describes the quantity calculations provided in PharmaSuite. Quantity calculations are used in the order explosion and in the Dispense and Weighing phase building blocks.

Global Unit of Measure Conversions

PharmaSuite and its underlying platform FactoryTalk ProductionCentre support the definition of global unit of measure conversion factors within the same physical quantities (e.g. length, time, mass, temperature). For conversions between different physical quantities, material-related conversion factors can be defined.

Global unit of measure conversion factors (e.g. 1 kg = 1.000 g) must not be misused to define non-global conversion factors that might even contradict already defined global conversion factors. This leads to non-predictable unit of measure conversions during execution.

If additional standard conversions need to be loaded into the system, this has to be handled by a system integrator.

Material-specific Conversion Factors

Updated for PharmaSuite 9.1.

Suppose there is a unit conversion from **kg** to **l** with a factor of **1.2** (and no offset). Converting **1 kg** to **liters** will result in **1 l**, whereas converting **1.0 kg** will yield **1.2 l**. The conversion function preserves the precision of the input.

A conversion offset is not supported by the Dispense, Inline Weighing, and Output Weighing phase building blocks of PharmaSuite.

Localized Signature Reason in the Production Execution Client

New for PharmaSuite 8.3

In the Production Execution Client (PEC), a reason is displayed for each electronic signature (e.g. Performed by, Reviewed by).

Since PharmaSuite 8.3, the reason for electronic signatures to be performed in the Production Execution Client can be localized. However, a localized reason must not exceed 80 characters.

The localized text is only used as a default during recipe authoring, i.e. the reasons used during execution are based on the configuration within the recipe.

Only if no reason is maintained in the recipe, the system defaults to the related message pack entry. If no message pack entry is maintained, the system defaults to the reason attribute of the related signature privilege.

Asian Locale and Exceeded Input Field Length

If an edit field has a length restriction (maximum N characters), the restriction does not work for Unicode fonts (e.g. Asian fonts), but arbitrary long strings can be entered into the field. In the end, the data cannot be stored to the database or be displayed correctly. This is an issue within Java and has not been solved yet (as of Java 6).

The system cannot assure the correct length of a given string automatically. However, in case the length restriction is exceeded, the string is no longer displayed correctly. Therefore, when entering and saving of data, the user needs to double-check that the system still displays the data correctly.

Equipment and Work Center Management

Equipment Entity's Context Information within Data Manager

New for PharmaSuite 8.2

If an equipment entity is used in the context of a workflow, PharmaSuite does not record the current/previous context in the **Context** tab of the equipment entity in Data Manager - Equipment. The context is still available in the equipment logbook (if maintained).
Only the order-related context is recorded in the **Context** tab, i.e., the previous order context information remains unchanged during the execution of a workflow until the equipment entity is used again in an order context. Additionally, the context is not reset if the equipment entity is used in the context of any workflow.

The evaluation of cleaning rules (e.g., during a subsequent weighing step) is based on the current/previous context that is presented in the **Context** tab of an equipment entity in Data Manager - Equipment. The previous order context now remains unchanged during the execution of any workflow (e.g., a cleaning workflows), including the information of the previously involved material (in the context of an order). That means, after the execution of a cleaning workflow (e.g., a minor cleaning), the previous order context information is still available for evaluation of pre-configured cleaning rules (e.g., **Minor cleaning required** vs. **Major cleaning required**) in the context of a sub-sequent weighing step of an order.

If an equipment entity (e.g., room, container) is exposed to other material while it is used within a weighing campaign (e.g., usage of a PharmaSuite workflow for Weighing without order or even weighing without any usage of PharmaSuite), an operational procedure (SOP) needs to be in place to ensure that the cleaning status of the equipment entity is set to an appropriate cleaning status (e.g., **Major cleaning required**). Thus, the equipment entity cannot be used again within the campaign without being cleaned appropriately.

Change Control within Data Manager

Last updated for PharmaSuite 8.4.

Data Manager allows to manage the following equipment-related master data: property types, equipment classes, equipment entities, and equipment graphs. In addition, work centers and stations are also maintained in Data Manager.

Change control is available for equipment classes, equipment entities, and equipment graphs. Best practice usage of the system only allows equipment classes and entities to be used during execution, if they are in an **Approved** status. However, in support of certain use cases (e.g., equipment qualification), the system allows to configure a recipe or workflow in a way, that equipment data can be used during execution, even if not in an **Approved** status, thus potentially being subject to change.

As a consequence, any change to data of objects that have already been used has to be applied very carefully in order not to break already approved recipes, workflows, or custom building blocks.

In addition, PharmaSuite also allows equipment classes to be moved back from the **Approved** to the **Draft** status. Any update of an equipment class that already has been approved earlier may have a severe impact on existing master recipes or master workflows. In order to support a related risk assessment, PharmaSuite provides a usage list for classes to give an overview of where a class is referenced as an equipment requirement in recipes, workflows, or building blocks.

For work centers and stations, no system change control is in place yet.

To separately manage the write or read access to Data Manager, the system provides two different access privileges for each of the two modes (Data Manager - Equipment, Data Manager - Work Center).

Expiry Date of a Current Equipment Status

New for PharmaSuite 8.1.

With PharmaSuite 8.1, status graph management was introduced in Data Manager to model specific status graphs for equipment entities (e.g., test status graph of a scale, cleaning status graph of a room).

For each status graph of an equipment entity, the system provides a **Current status** property and, if a status is configured to expire, an **Expiry date** property.

The **Expiry date** always represents the expiry date of the **Current status** only and needs to be set upon the latest status transition. This means, only the current status can expire automatically.

In case a specific use case requires the management of an expiry date other than for the current status of a graph, this needs to be modeled by managing separate expiry date runtime properties in addition to the **Expiry date** property provided by the system.

Forcing an Equipment Status in Data Manager

Last update for PharmaSuite 9.2.

With PharmaSuite 8.1, status graph management was introduced in Data Manager to model specific status graphs for equipment entities (e.g., test status graph of a scale, cleaning status graph of a room).

Since PharmaSuite 8.2, Data Manager provides an operation that allows to change runtime data, such as runtime properties, graph-related data (e.g., status, expiry date), and grouping situations, of an equipment entity in a read-only status. Since the **Current status** property of an entity is always modeled as a runtime property, this operation allows to force a status of any given status graph of an entity.

Typically, any equipment-related status transition is triggered by a recipe or workflow on the shop floor. Per configuration in the status graph of an equipment object, additional actions can take place automatically once a specific status transition has been triggered (e.g., increase of a counter).

For those exceptional cases when an equipment-related status needs to be forced in Data Manager, it is important to understand the following implications:

- Forcing a new status in Data Manager by updating the **Current status** property value does not represent a status transition as defined within the related status graph. Only the related runtime property is updated.
- As a consequence, none of the actions are performed automatically that can be defined within a status transition of the related status graph and its associated target status.
- Any related action (e.g., increase of a counter or re-calculation of an expiry date) that needs to be executed from a use case perspective requires a manual update of the related runtime property.

The main reason for this solution concept is the fact that use cases that require to perform a forced status change in Data Manager can be very different. Depending on the exceptional situation on the shop floor, further runtime properties may need to be updated or not. This shall always be an explicit action based on a conscious user decision rather than an implicit action that is performed automatically by the system.

Since PharmaSuite 9.2, Data Manager additionally provides a functionality to trigger an equipment graph status transition with all its defined actions on a specific single entity or on grouped entities.

Export/Import of Entities and their Class Assignments

Assuming an entity A is assigned to a class B in the source system.

If entity A is exported and imported into a target system that has no class B available yet, the assignment is automatically deleted upon the import of entity A.

However, if later on, class B is exported and entity A is still assigned to class B in the source system, the assignment will also be re-established in the target system upon the import of class B.

Export/Import of Entities, Classes, and Graphs in View of Change Control

Updated for PharmaSuite 8.1.

Change control is available for equipment classes, equipment entities, and equipment graphs. However, the status and status history information of a class, entity, or graph from its source system is not part of the export/import process. As a consequence, classes, entities, and graphs always need to be approved in the target system, independent of their previous status in the source system.

During the import of classes, entities, and graphs, specific rules apply in case the objects already exist in the target system. In general, update of an object with the import function is only allowed if the object in the target system is in the **Draft** status.

- A class, entity, or graph can only be imported, if it is in the **Draft** status in the target system.
- If an object with dependent objects is imported (e.g., class with assigned entities, entity with assigned graphs), specific rules apply. For more information, please refer to "Functional Requirement Specification Data Management" [A15] (page 45).
  Generally spoken, if the dependent object's content in the target system differs from the source system (which means it will be updated along with the import), the **Draft** status is required in the target system.
  If the dependent object in the target system is already in the **Verification** or **Approved** statuses, but its content is identical in the source and target systems, it will simply be skipped during the import and the import of the parent object will be successful.

OPC-supported Data Type Mappings

Updated for PharmaSuite 9.1.

PharmaSuite only supports the explicitly defined data type mappings listed in the table below:

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

OPC standard data type	Description	Common OPC names	Mapped to Live Data data types	Mapped to PharmaSuite technical property
VT_UI2	2-byte unsigned integer	**USHORT**	**Integer**	BigDecimal
VT_UI4	4-byte unsigned integer	**ULONG**	**Integer**	BigDecimal
VT_ARRAY	Arrays	ARRAY OF ...	not supported	N/A

When defining the tag name (tag path) of numeric equipment entity properties in Data Manager - Equipment, it is crucial to ensure that the OPC data type behind the defined tag name matches the pre-defined Live Data data type of the property, which was defined on property type level.

PharmaSuite supports the check with a manual **Verify** function. It double-checks the validity of the tag name and the correct data type mapping. In case the check fails (name or data type), PharmaSuite displays a warning message that the given tag name is not valid.

Note: If PharmaSuite runs in an environment without a connection to the OPC server, an equipment entity can be saved even if tag names are not defined properly, or data types do not match.

Example of the consequences when a numeric property with a wrong data type mapping is used during execution:
A **Get OPC value** phase is configured to support the **Integer** Live Data type, but the related tag name is mapped to a **REAL4** OPC data type (Float). During execution, while reading from the tag, an arbitrary conversion may take place and provides just the **Integer** portion of the value (25.678 in the PLC is displayed and recorded as 25 in PharmaSuite). There is no indication to the operator that a conversion took place.

Locale-specific Logbook Entries

PharmaSuite writes equipment-related logbook entries as a string according to the actual Locale (e.g., English-US). This might have an impact on the way how expression conditions (e.g., related to a transition within the recipe graph) have to be defined.

If string-based information from the logbook is used to define expression conditions, this results in expressions that now also depend on the Locale of the running system. As a consequence, the recipe can only be executed on a client that is running based on the same Locale compared to the recipe definition.

Export/Import of Work Centers and Stations

Data Manager - Work Center, introduced with PharmaSuite 5.2, allows to manage the following master data: work centers and stations.
Currently there is no export/import function for work centers and stations available within Data Manager.
To transfer these objects from system A to system B, Process Designer's DSX export/import must be used. See also "Work Centers do not export their Production Line(s)" (page 38).

Registration of Devices at Stations

Updated for PharmaSuite 8.3

Since PharmaSuite 5.2, the system supports certain agility use cases that allow to run the Production Execution Client and the related recipe or workflow processing on multiple stations per work center and also on multiple devices per station. One specific new feature in this context is the **Register at Station** function, which supports the usage of mobile devices. It allows an operator to register his current device at another station, provided the operator is authorized to execute recipes on that specific station.

Before PharmaSuite 5.2, the setup of work centers, stations, and devices was always based on two facts:

- There is a 1:1:1 relation between those artifacts.
- The related setup cannot be changed by an operator in the Production Execution Client.

Given the new agility use cases that are now supported with PharmaSuite 5.2, there are certain aspects that need to be considered during the initial setup of work centers, stations, and devices, and also during the configuration of the Production Execution Client running on those devices and stations.

- In order to control which user is allowed to log in at which station, assign respective access privileges to each station in Data Manager.
- In order to not allow a registration of a stationary device at all, disable the manual and scanner-based execution of the **Register at Station** function for the given station.
- In order to enforce that a registration of a mobile device can only be done with the barcode right at the new station's location, disable the manual execution of the **Register at Station** function (or only allow it for a supervisor) and only allow a scanner-based registration for the related stations.
- In order to control the correct status of equipment required for order execution, consider to model explicit steps for equipment identification in a master recipe.

For work centers of the **Dispense** type, there still is a 1:1:1 relation required between work center, station, and device.

Automation Integration

Automation Infrastructure-related Data

For each equipment entity, the relevant Automation Integration (AI) server name and the Live Data Area path can be maintained within Data Manager - Equipment.

In case the AI server name or the Live Data Area path are changed for a specific equipment entity after it has already been used on the shop floor, the respective Production Execution Client (PEC) needs to be restarted in order to reflect the changes.

Limited Real-time Capability

PharmaSuite is not designed to replace a real-time SCADA system, but to ensure compliant manufacturing from an MES perspective.

Two of the key advantages of the MES automation integration are the ability

- to automatically read, write, and monitor values of equipment properties and document the result in the batch report, and
- to record exceptions if expected values or ranges are violated or communication failures occur.

Usually, the required technical infrastructure works fine and these capabilities can be considered to provide "semi-real-time" results. However, due to the nature of the technical setup, there still is a risk that the communication between MES and the automation layer may be delayed or may fail completely.

During recipe or workflow execution, PharmaSuite provides an exception handling that allows to respond to this kind of communication failures (e.g., record the results manually) and to record the exception in the batch report.

Limitation of Monitoring Capability

The **Monitor numeric value** phase, introduced with PharmaSuite BB Delta, provides the capability to monitor a value-related tag, based on a pre-defined condition and monitoring duration (e.g., for the next 5 minutes monitor if the value exceeds 23 °C).

However, one specific tag can only be monitored against one condition at a time. That means, the system does not support to monitor one tag simultaneously against different conditions (e.g., phase 1 monitors T1 > 23 °C, phase 2 monitors T1 > 50 °C), not even if different **Monitor numeric value** phase instances are used or the phases are run on different clients.

To monitor different conditions simultaneously, the automation layer needs to provide different tags for the same data to be monitored.

Equipment Automation Tag Changes

New for PharmaSuite 10.02.00

Since PharmaSuite 10.02.00, all tags of an entity ("Tag name", "Indicator tag"), defined with a path, form the tag group of this entity together. The initial creation of the tag group is done with the first successful tag verification or the first read/write operation. If an equipment entity shall be used in a not approved status, changes on entities tag definition can be populated by a further successful tag verification.

Dispensing

Replacing Sublots in the Production Management Client

New for PharmaSuite 8.3

In general, as exceptional use cases can vary in a wide range, a key idea of PharmaSuite to handle exceptional cases is to provide as much flexibility as possible, in order to record the corrective actions properly in the system. However, a high level of flexibility always implies that there is less user guidance in terms of automated calculations or checks by the system.

An example of an exceptional use case that has to be reviewed carefully and typically signed by a supervisor as a witness (double signature) is the replacement of a sublot and related quantities in the Production Management Client.

The system provides flexible ways to resolve a wide range of replacement-related use cases. For example:

- The Production Management Client allows to replace a single active substance sublot even though the compensation material has already been dispensed completely.
  (This is restricted in the Production Execution Client. There, all compensation material has to be replaced first, before active material can be replaced.)
- The Production Management Client allows to adjust the new default remaining planned quantity of a material by the user, related to each sublot to be replaced (planned quantity can be edited.).
- Even after all of the relevant sublots were marked as replaced and as long as processing of the reactivated order step has not been started yet, the new remaining quantity of the material item can still be adjusted in the Production Management Client with the **Increase quantity of order step input** action (1st OSI split position).

In other words, PharmaSuite automatically calculates a resulting remaining quantity for each individual sublot as a default planned quantity (for active materials, the remaining quantity is recalculated according to the actual/planned potency of the replaced sublot). However, the quantity has to be reviewed and adjusted manually by the user as needed.

This is especially true in case the entire set of sublots of an active or compensation material position is replaced. Due to potential rounding errors (active material) or previous compensator calculations (compensation material), this typically requires a manual quantity adjustment in order to reset the remaining quantity to the original planned quantity of this material position.

Tracking of Exceptional Events for Order/Workflow Steps in the Production Management Client

Updated for PharmaSuite 9.1

As part of the **Manage Orders/Workflows** use case in the Production Management Client, PharmaSuite provides capabilities to perform the following operations:

- Define an alternative material (order step input)
- Edit the comment of an order/workflow
- Edit the comment of an order/workflow step
- Dispatch order/workflow steps
- Allocate batches
- Add material-related comments for non-Dispense order steps

All of these operations can be controlled by respective privileges. Since PharmaSuite 9.1, an explicit **Order and Workflow Changes** table is used to track those operations in the database and document the changes in the batch or workflow report.

When an order or workflow changes its status from **In process** to **Finished**, PharmaSuite checks if order or workflow changes have been tracked. If so, the system automatically records an "order/workflow changes"-specific exception for the last unit procedure.

Cleaning Rules and the Room’s Context Information within Data Manager

New for PharmaSuite 8.2

If an equipment entity is used in the context of a workflow, PharmaSuite does not record the current/previous context in the **Context** tab of the equipment entity in Data Manager - Equipment. The context is still available in the equipment logbook (if maintained).
For more information, please refer to "Equipment Entity’s Context Information within Data Manager" (page 8).

Tolerance Calculation

Updated for PharmaSuite 10.0.

In versions of PharmaSuite prior to 3.1, during dispensing, the system reduced the tolerance range of a weighing item by the scale resolution **d** (e.g., 0.1 g) per split position.

Since PharmaSuite 3.1, the tolerance range is no longer reduced by the system. This functionality has been removed in order to:

- support a lower or upper limit of 0% and
- avoid issues related to restricted tolerance ranges where a weighing position cannot be completed anymore after multiple split positions have been created and no tolerance range is left.

This behavior has to be taken into consideration when tolerances are defined within the master recipe.

Since PharmaSuite 8.0 (corrected in the **Select scale** phase building blocks of PharmaSuite 10.0), the tolerance calculation has been changed slightly, based on the following assumptions:

- The **tolerance** definition within a recipe rather has the meaning of an **accuracy** or **precision** definition of a quantity, which is reflected by the question "What is the maximum/minimum value of the real weight that I will accept, not knowing yet which scale resolution will apply during execution?"
  For this reason, we do not allow lower AND upper tolerances to be 0 at the same time, because there is no physical scale available with a precision of +/- 0.
- An update is applied with PharmaSuite 8.0 that recalculates the nominal tolerance range during execution based on the resolution of the used scale:
  [roundUpAccordingScaleResolution(lower tolerance + resolution) ..
  roundDownAccordingScaleResolution(upper tolerance – resolution)]
  In this way, it is assured that weighing **In tolerance** during execution results in a physical weight that really is within the original tolerance definition of the recipe.
- The algorithm within the **Select scale** phase that identifies suitable scales follows the same approach. Only scales are allowed with a configured weighing range that covers, for the planned value, a valid nominal tolerance range (calculated as described above) or the resolution of the scale is sufficient in case the planned value is outside of the scale's weighing ranges. A valid weighing range still can be reached due to the additional tare or an underweight situation. The resolution is sufficient if the scale's resolution value is smaller than or equal to twice the tolerance range.

Only Identification

Updated for PharmaSuite 8.4

If **Only identification** is the default weighing method and no other weighing methods are allowed according to the master recipe, an operator on the shop floor will not be able to handle overweight situations other than through another **Only identification** action.

In case a final dispensing step will be required on the shop floor (e.g., **Net** weighing), a reasonable weighing method must be allowed within the master recipe and the required weighing phase building blocks (for scale selection, taring, and weighing) must be part of the recipe definition.

Scale Timeout

After a communication timeout, the tare value presented in the Dispense application may temporarily show a different tare value than the display of the scale itself.

If the scale is unstable when it receives the tare command, it will not respond until eventually the scale gets stable and the tare can be determined. There is no timeout at the scale itself and the tare command is not discarded. The Dispense application terminates waiting for the tare response from the scale after a timeout (10 sec.), displaying an error message and remaining in the **Tare** phase.

The scale will tare itself (switch from gross to net display) immediately after the stabilizing (even minutes later). However, this is not fully reflected in the representation of the tare value in the Dispense application: the net value is shown instead of the tare value.

In order to correct this, the operator has to repeat zeroing and taring until the scale display (of the scale) and the representation within the Dispense application are synchronized again.

Precision Check for Manually Entered Weighing Values

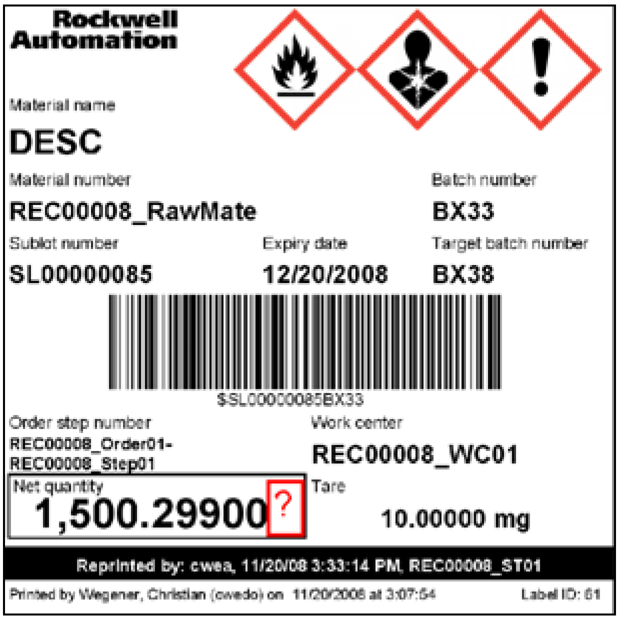
New for PharmaSuite 10.0

In case the value of a weight is entered manually since the scale is configured as a manual scale or the scale is offline, the precision of the entered value must match the resolution of the used scale.
If the scale is configured as a multi-range scale, the resolution of the appropriate range is used to verify the manual input. The entered value must be a multiple of the range resolution.

Labeling and Reporting

Size of Value Fields

If the numeric value of the net quantity has too many digits, the Unit of Measure may be truncated.



Rockwell Automation

Material name

DESC

Material number

REC00008_RawMate

Batch number

BX33

Sublot number

SL00000085

Expiry date

12/20/2008

Target batch number

BX38

$SL00000085BX33

Order step number

REC00008_Order01-REC00008_Step01

Work center

REC00008_WC01

Net quantity

1,500.29900?

Tare

10.00000 mg

Reprinted by: cwea, 11/20/08 3:33:14 PM, REC00008_ST01

Printed by Wegener, Christian (cwedo) on 11/20/2008 at 3:07:54

Label ID: 61

Figure 1: Example sublot label

Solution: As part of the project-related fit-gap-analysis, the required number of digits needs to be analyzed. If required, sublot labels and the dispensing report need to be adapted.

Number of Printouts

Updated for PharmaSuite 7.1.

In the pre-defined **Material receipt** and **Sublot split** workflows of the Production Execution Client, when printing sublot labels, the operator can define the number of labels to be printed per sublot and whether the printed labels will be collated or will not be collated.

However, currently the **Number of labels per sublot** option is **read-only**. Thus, the operator cannot change the default value. This is due to the fact that the system currently does not track the number of labels that were printed as part of one print job. (Separate reprint print jobs are tracked.)

Nevertheless, in case the customer wants to use this functionality despite the fact that the number of labels is not tracked, the **Number of labels per sublot** option can easily be enabled by a system integrator.

Barcode Support

Updated for PharmaSuite 8.4.

The default barcode type in PharmaSuite for sublot labels is Code 128. However, during execution (e.g., material receipt, batch creation), there is no automated check of an operator’s data input against conformity with the barcode type.

As a consequence, an operator could type non-barcode-conforming letters as part of an identifier (e.g., "Ä", "Ö", "Ü" within a batch identifier), which afterwards would lead to a system error during barcode identification (the barcode represents a combined identifier from sublot and batch).

An operational procedure (SOP) needs to be in place to ensure that in case of barcode-relevant identifiers, only those identifiers are defined and used, which conform to the used barcode type.

Localization of Reports

When localizing a report, be aware of differences in text length in the target language. If a string does not fit into a report field, JasperReports leaves the field empty instead of displaying an ellipsis or even a truncated string. This means that text expansion or only a different time/date format can become an issue and requires the report layout to be changed.

Recipe Management

Material Master Data

Updated for PharmaSuite 10.0

Material master data is not under version control. As a consequence, data might be changed by an authorized user at any time.

Material master data is under audit trail.

Appropriate access privileges and an operational procedure (SOP) need to be in place to ensure that critical data such as a **Material UoM conversion** is not updated in the system.

Positions of BOM Items

In general, the system supports alphanumeric BOM item positions.

However, if one of the BOM items is of the **Filler substance** weighing material type, each BOM item position must be numeric (leading zeros might be included).

Scalability of Master Recipes – Expected Number of Produced Sublots

New for PharmaSuite 7.1.

In general, recipes are scalable in PharmaSuite. In case the planned quantities of an order and its underlying master recipe are different, during order explosion the system automatically applies a related scaling factor to the planned quantities of all material parameters of this order (if not marked as **fixed quantity**).

However, for Output Weighing, the automated scaling of quantities does not apply to the **Number of sublots** process parameter, which is used to configure the expected minimum and maximum number of produced sublots.

In order to support scalable recipes, the respective calculations need to be modeled explicitly for the process parameter with an information flow function (e.g., base value multiplied by the **Order Scaling Factor** function).

Recipe and Workflow Management

Room Status and Room ID Functions

New for PharmaSuite 8.1.

Prior to PharmaSuite 8.1, a room was always bound implicitly to a running unit procedure, based on the work center assignment in its master data and the current work center of the running Production Execution Client. An example that is still valid in PharmaSuite 8.1 is the implicit binding of a room along with the identification of material in Dispense or Inline Weighing with the **D Identify material** phase.

The Expression editor of Recipe and Workflow Designer provides the Room Status and Room ID functions. Both are evaluated in the same way during execution, based on the work center assignment of the room and the current work center of the running Production Execution Client.

With PharmaSuite 8.1, rooms can also be bound explicitly with the **Identify equipment** phase, independent of the work center assignment of the room equipment entity in Data Manager - Equipment.

An operational procedure (SOP) needs to be in place to ensure that if in such cases the Room Status and the Room ID functions are used within a recipe, only rooms can be identified explicitly whose assigned work center is also the current work center of the running Production Execution Client.

Alternatively, the current room status can be retrieved with the Equipment graph status (runtime-related) function (equipmentGraphStatus(arg1, arg2 [, arg3]). With an equipment object as first argument (retrieved with an output variable of the **Identify equipment** phase), a purpose as second argument, and an attribute identifier (KEY or DISPLAY) as optional third argument. In this way, the current status is always retrieved from the object that has been explicitly bound to the current unit procedure.

Definition of Signature Privileges for Exceptions

New for PharmaSuite 8.1.

For each signature privilege that is assigned to a specific phase during recipe/workflow authoring, a related risk level has to be defined. According to the risk assessment for each individual exception, the signature privilege with the same risk level will be required in order to sign an exception during execution.

With PharmaSuite 8.1, the following new risk levels were introduced: **Low (mandatory comment)**, **Medium (mandatory comment)**, and **High (mandatory comment)**. If one of the new risk levels is defined during recipe/workflow authoring, a mandatory comment has to be provided when an exception is signed during execution.

Please note that the risk levels without and with a mandatory comment (e.g. **High** and **High (mandatory comment)**) represent different risk levels to which different signature privileges must be assigned. This means, a signature privilege with risk level **High** does not apply to an exception with a risk level **High (mandatory comment)**.

Loops Built into EBR Recipes

Last updated for PharmaSuite 9.0.

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

However, SFC is limited as follows:

- Loops are not allowed for process steps that are directly located after or before a branching.
- Loops are only supported for phases and operations, whereas parallel and XOR branching is supported for phases, operations, and unit procedures.

These rules need to be considered during recipe and workflow building.

Transition Conditions within SFC Modeling

Updated for PharmaSuite 8.4.

When defining transition conditions within an SFC model, ensure that the conditions are consistent and unique.

- Unless all of the potentially required values are covered (logical gap), the system may run into a dead end during recipe or workflow execution.
- If definitions are not unique (logical overlapping), the system may respond in a non-determined way during recipe or workflow execution.

If the system is not able to evaluate transition conditions and the recipe appears to hang, a specific recovery capability is available in the Production Management Client since PharmaSuite 8.4 (see section "Recovery Capabilities to Resolve Issues during Execution" (page 30)).

NOT Operator Used in Expressions

PharmaSuite interprets the expression **NOT A OR B** as **(NOT A) OR B**, but not as **NOT (A OR B)**.

This interpretation is in line with most programming languages (e.g., Java, SQL).
The same interpretation applies to the combination of the NOT and AND operators.

Limited Support of Dynamic Expressions in the Context of Get Values and Show Values Phases

Updated for PharmaSuite 10.0.

In general, the configuration of process parameters as part of recipe authoring is supported by the information flow mechanism, which includes the ability to define dynamic expressions.
This also applies to the process parameters of the PharmaSuite **Get values** and **Show values** phases introduced with PharmaSuite 6.0 in support of IPC-related data collection and data representation use cases.

However, some limitations apply if the **Show values** phase is used to present data that previously has been collected by the **Get values** phase.
In this case, the system only supports dynamic expressions as long as the result of each expression during execution (at runtime) is constant within a given process order/control recipe. This applies to the following phase, process parameter, and attribute:

- Get values / <Master (bundle identifier)> / Short description
  (used for table headers in the **Show values** phase).

Significance of Change Requests

Updated for PharmaSuite 7.1.

The purpose of a change request is to enable the user to replace specific custom building blocks automatically and, depending on the defined target status (of the recipe or workflow), to directly create new versions of recipes or workflows in a **Valid** status. In this context, failures in defining the scope of a change request might result in corrupt recipes or workflows that are available for production, which means there is a direct impact on product quality.

Therefore, each change requests must undergo its own approval cycle and conscious risk assessment has to be applied during review and approval.

Interruption of a Change Request

PharmaSuite allows to interrupt the execution of a change request in a controlled way by using the Cancel function. If the user confirms the interruption, the execution of the change request is canceled. If not, the execution can be continued.

Any other situation that leads to an interruption of the execution of a change request must be avoided, as this most likely will lead to inconsistent data. This applies, e.g., to situations when the execution of a change request is running for several hours and the PharmaSuite client is automatically moved into a stand-by or hibernate status while the execution is still running.

Handling of Custom Building Blocks Bundle Parameters within Smart Replace or Change Request

New for PharmaSuite 10.0

Prior to PharmaSuite 10.0, bundle parameters added within a used custom building block to a recipe or a custom building block were removed when the custom building block was replaced with smart replace or by a change request. This system behavior is changed as follows: a bundle parameter always remains at the old element if it is not part of the new element. Removing a bundle parameter with smart replace or by a change request is not possible. The deletion of a bundle parameter is a manual activity.

Building Blocks with Empty Exception Texts

New for PharmaSuite 10.0

With PharmaSuite 10.0, the execution of a phase building block will no longer be blocked due to a missing exception text for a user-triggered or post-completion exception. Instead, a default text will be added to the exception. The localizable default text can be changed (message pack: pec_ExceptionMessage, message ID: defaultExceptionText_Msg).

Order Management

Cancel Order or Abort Order Step

Last updated for PharmaSuite 8.4.

Changing the status of an order or workflow to **Canceled** or aborting an order or workflow step (only available for non-Dispense order steps) is a highly sensitive task that needs to be considered carefully. It is assumed that situations on the shop floor are cleaned up first, before an order/workflow is canceled or an order/workflow step is aborted in the Production Management Client.

However, in case an order/workflow step on the shop floor is still **In process** while **cancel order/workflow** or **abort order/workflow step** is performed in the Production Management Client, the system displays a message at the related Production Execution Client to inform the operator about the canceled order/workflow or aborted order/workflow step, respectively. Finally, the system automatically changes the status of all related order/workflow steps from **In process** to **Canceled** and the related recipe or workflow elements (unit procedure, operations, phases) are aborted.

Depending on the situation on the shop floor when the order/workflow is canceled or the order/workflow step is aborted, sublots, work centers/rooms, and equipment objects may still be bound to the then aborted order/workflow step.
In order to clean up the related data in the system, sublots can be released by a system administrator in the Production Management Client.

Runtime property values and graph statuses of equipment objects (e.g., scales, rooms, containers) can be cleaned up in Data Manager, if required. In case an equipment object is still bound to a canceled or aborted order, the **Force unbind** function can be applied to the equipment object.

Finish Order with Aborted Order Step

New for PharmaSuite 8.4.

Since PharmaSuite 8.3, the system allows to finish an order with order steps that are in an **Aborted** status, as long as these order steps were successfully completed at least once.

This feature supports the exceptional use case of an already completed order step (unit procedure) that has been reactivated accidentally, i.e., the reactivated instance shall not be executed.

In such a case, the reactivated instance of the unit procedure first needs to be started in the Production Execution Client but does not need to be executed. The then running unit procedure instance can be aborted right away.

In case all order steps of the order have been completed at least once, this order with order steps in the **Aborted** status now can manually be set to **Finished** in the Production Management Client and reviewed in the Exception Dashboard of the Production Response Client.

Recipe Processing

Prorated Quantities of Material Parameters during Execution

New for PharmaSuite 7.1.

With PharmaSuite 7.1, prorate factors were introduced. They can be applied during execution in order to automatically reduce the planned quantity of input and output materials of a current unit procedure, based on the yield calculation of produced output materials from one or more preceding unit procedures.

The definition of prorate factors during recipe authoring and the handling of override exceptions during execution need to be exercised with reasonable care. For example, certain aspects need to be considered if multiple material inputs of one unit procedure are split across multiple Inline Weighing operations and phases:

- The recipe author has to define the same prorate factor for all of the relevant phases, in order to ensure that during execution all of the unit procedure-related materials are prorated by the same factor.
- During execution, the prorate factor can be overridden with the **Override prorate factor** user-triggered exception, which only applies to the materials that are assigned to the specific phase. Again, if material inputs are split across multiple Inline Weighing operations and phases, the **Override prorate factor** exception needs to be performed again for each of the relevant phases.

For more information, please refer to "Functional Requirement Specification Dispense and Inline Weighing" [A12] (page 45).

Handling of Runtime Properties during Exceptional Weighing Situations

New for PharmaSuite 8.1.

The Dispense and Inline Weighing phases and the Output Weighing phases provide specific exceptions that have an impact on the involved equipment objects. In general, the current status and/or context information is reset when such an exception is recorded, e.g. a **Return to material identification** user-triggered exception.

However, the value of a runtime property is never reset automatically by the system when an exception is recorded, because most likely this also requires further process steps on the shop floor. Runtime properties, in case they have to be reset manually as a consequence of such an exceptional situation, always need to be updated by a data administrator in Data Manager (or alternatively along with the execution of a respective clean-up workflow on the shop floor, if enabled by project-specific phases).

For more information, please refer to "Functional Requirement Specification Dispense and Inline Weighing" [A12] (page 45) and "Functional Requirement Specification Output Weighing" [A16] (page 45).

Handling of Loaded Scales during Inline Weighing

New for PharmaSuite 8.1.

During preparation of a target container in Output Weighing, the **Weigh** phase can be used to mark the used scale as currently loaded (scale's runtime property of the **Current Load (RS)** purpose). This results in skipping the zeroing of the scale (with the **Select scale** and **Get weight** phases) and skipping its release check (in the **Release scale** phase) during subsequent process steps.

Especially during Inline Weighing, the scale's runtime property of the **Current Load (RS)** purpose also needs to be evaluated as part of certain transition conditions.

For more information, please refer to section "Transitions within an Inline Weighing Operation" in "Functional Requirement Specification Dispense and Inline Weighing" [A12] (page 45).

Termination of Order Steps

Update for PharmaSuite 9.0.

If the **Abort-and-reactivate-enabled** capability for a unit procedure has been configured in Recipe Designer, the related running order step can be aborted by the operator and a new reactivated instance will be available to be processed at the same or a different work center.

If the **Abort-and-reactivate-enabled** capability for a unit procedure has not been configured in Recipe Designer and thus is not available in the Production Execution Client for this running unit procedure, an abort and reactivate can still be achieved for this specific order step by an authorized user with the **Manage Batch Orders** use case in the Production Management Client. This will result in an exception that is automatically added to the related unit procedures.

Recipe and Workflow Processing

NULL Handling of Quantity-related Information Flow Functions

New for PharmaSuite 7.1.

With PharmaSuite 7.1, the following information flow functions that allow to access MFC position-related quantity data were introduced:

- Planned Quantity - Original function: originalPlannedQuantity
- Planned Quantity - Execution function: plannedQuantity
- Actual Split Quantity function: actualSplitQuantity
  Returns the actual recorded quantity of the last closed target (not available for output materials).

- Total Actual Quantity function: totalActualQuantity
  Returns the total recorded quantity of a material position minus replaced quantity or quantity that was declared as waste.

Usually, an accurate proper recipe design ensures that the quantity-related functions do not return NULL during execution, i.e. quantities already have to be recorded in the database before they are referenced within an information flow function.

However, in case the functions retrieve NULL from the database, this is handled differently by the system with respect to the values that the functions actually return during execution.

- Planned quantities:
  If no planned quantity is maintained (e.g. **Planned quantity mode** is set to **None**), the related functions also return NULL during execution.
  If a planned quantity-related function is used in a context without a defined planned quantity, this is considered to be a flaw in the recipe design.
- Actual split quantity (for input material) and Total actual quantity (for input and output materials):
  If no actual quantity is available in the database (e.g. no material has been weighed or produced yet), the related functions do not return NULL during execution, but the "0 <UoM>" value.
- Actual split quantity (for output material):
  As this function is not supported for output materials, it always returns NULL during execution if it refers to an output material position.

Update of Specific Context Data

Updated for PharmaSuite 7.1.

As part of the information flow capabilities, specific context data can be made available for phases during execution. This includes order context details, MFC position context details, and location context details.
For a subset of the available data, an update of that master data is not populated automatically to the still running Production Execution Client and the EBR server. This limitation only applies to the following data, which usually is not subject to change:

- Order context: Produced material short description and material type.
- MFC position context: Material short description and material type.
- Location context: Room-related storage area and work center information.

In order to have any update of the listed data being reflected on the shop floor, both the Production Execution Client and the EBR server need to be restarted.

PEC, EBR Server, and OES Do Not Refresh Cached Data Automatically

Issue: As an example, equipment master data is cached on the Operation Execution server starting with its first usage. The cache is only refreshed when the OES is restarted or if the business logic of a phase explicitly refreshes the cache. This means that updates to already cached master data may not be available for business logic that runs on the OES unless the server is restarted.
Refers to FTPS-3973.

Solution: An operational procedure (SOP) needs to be in place to ensure that master data is not updated while these objects are used during execution.
In case master data needs to be updated, an explicit refresh of the cache can be triggered as follows:

- Restart the OES.
- Restart the EBR server.
- Restart the Production Execution Client, perform the **Change User** action, or the **Change Station** action.

Interruption of Order or Workflow Steps - Resume at the Same Work Center

Updated for PharmaSuite 7.1.

If a running order or workflow step needs to be **interrupted** for being **resumed at the same work center**, the operator first needs to finish or detach the still running operations before he can exit the system. The operation will be available to be resumed the next time the operator logs in at the same work center again.

For non-Dispense operations, in case the **Detachable** capability has not been configured in Recipe Designer for an operation and thus is not available in the Production Execution Client for this running operation, a detach can still be forced for this specific operation by an authorized user within the Production Management Client (**Unlock objects** function). This will result in an exception that is automatically added to the still running phase or phases of this operation.
This also applies to Inline Weighing and Output Weighing operations.

In case of a Dispense operation, the operator needs to execute a **Return to material identification** exception first and then to stop processing the order step by confirming the phase before exiting the system. For more information, please refer to the **BB complete (SR0200.2.3)** function in "Functional Requirement Specification Dispense and Inline Weighing" [A12] (page 45).

Interruption of Order or Workflow Steps - Resume at a Different Work Center

Updated for PharmaSuite 7.1.

If the **Detachable** capability for a unit procedure has been configured in Recipe and Workflow Designer, the related running order or workflow step can be interrupted by the operator to be **resumed at a different work center**. Before the unit procedure can be detached, the still running operations need to be finished or detached first.

If the **Detachable** capability for a unit procedure has not been configured in Recipe and Workflow Designer and thus is not available in the Production Execution Client for this running unit procedure, the current release of PharmaSuite does not offer any option to force a detach of a unit procedure/order step/workflow step in the Production Management Client. The only workaround is:

- For order steps: to complete or abort the unit procedure/order step, which then allows to reactivate the order step and to process it at a different work center.
- For workflow steps: to complete or cancel the unit procedure/workflow step and to start a new workflow at a different work center.

Room Status Check

Updated for PharmaSuite 8.1.

When a recipe or workflow is started in the Production Execution Client, the system supports an implicit room status check only within the **Identify material** phase for Dispense or Inline Weighing.

No other implicit room status checks are implemented. If required, explicit room status checks can be defined and configured by a recipe or workflow author with the **Identify Equipment** phase. Appropriate room-related requirements must be assigned to the phase.

Recovery Capabilities to Resolve Issues during Execution

New for PharmaSuite 8.4.

Since PharmaSuite 8.4, a toolset for administrators is available that provides certain recovery capabilities to resolve issues during execution that are caused by a faulty or undefined recipe parameter setup (e.g., information flow expressions of a process parameter or transition conditions that cannot be calculated by the system).

Any recovery action is a highly sensitive task that needs to be considered carefully. It requires an electronic signature and will be tracked as an exception in the batch record, according to the related risk and signature privilege definitions on configuration key level.

Abort Phase

New for PharmaSuite 8.4.

Aborting an active phase can be triggered in the Navigator view of the Production Execution Client, which results in a specific "Phase execution aborted" exception.

The following implications have to be considered for any aborted phase and a respective risk assessment has to be performed on a case-by-case basis:

- Aborting a phase bypasses all UI extensions (e.g. phase completion signature) and other phase-specific checks that may veto the completion.
- Execution of the phase-specific business logic will be incomplete. This may result in incomplete data having been recorded in the database.
- The information flow will be broken if any output variable of an aborted phase is used later on in the control recipe. An aborted phase already represents its own instance of the phase, however, any output variable of an aborted phase is undefined except for the output variables provided by the framework.
- After the phase has been aborted, processing of the operation continues normally, according to the SFC modeling in the underlying recipe.

Repair Phase

Updated for PharmaSuite 10.0.

Repairing an active phase can be triggered in the Navigator view of the Production Execution Client, which results in a specific "Phase repair mode started" exception and a "Phase parameters repaired" exception.

- Repairing is only available as long as no exceptions have been recorded for a phase. This restriction does not include "phase repair mode"-specific exceptions. Customer-specific implementations can further exclude specific exceptions from this rule.
- Repairing is not available for phases that are explicitly excluded from being repaired, according to the **PhaseRepairModeBlackList** configuration key. Currently, this applies to the **IPC Get values** phases with version <= [2.0].
- Note: For customer-specific implementations of phases, a risk assessment shall be performed in order to identify any phase that needs to be excluded from the repairing capability.

The following steps and implications have to be considered for any repaired phase and a respective risk assessment has to be performed on a case-by-case basis:

- The current phase instance is aborted automatically and the system records the "Phase repair mode started" exception that is assigned to the aborted instance of the phase.

- For the aborted phase instance, execution of phase-specific business logic will be incomplete. This may result in incomplete data having been recorded in the database.
- The system automatically creates a new instance of the phase. It is first displayed in the repair mode that allows to override its process parameters.
- Only the current numeric or string value of any process parameter, which might result from the calculation of an information flow expression, can be overridden. Expressions themselves cannot be defined or changed during the repair mode, which also includes object-related parameters (e.g. equipment object).
- Upon completion of the repair mode, the system records the "Phase parameters repaired" exception that is assigned to the new instance of the phase. As part of the exception, all changes to process parameter values are recorded.
- Finally, the new instance of the phase is displayed again with its regular user interface and can be processed as usual.
- After this phase instance has been completed, processing of the operation continues normally, according to the SFC modeling in the underlying recipe, and the output variables of this latest instance are available as part of information flow.

Force Execution Transition

Updated for PharmaSuite 9.0.

Forcing a stalled transition can be triggered in the **Manage Orders** use case of the Production Management Client, which results in a "force execution transition"-specific exception.

The following implications have to be considered for any transition to be forced and a respective risk assessment has to be performed on a case-by-case basis:

- For a given order/workflow, the system displays a list of stalled transitions that are related to XOR branches. It is in the responsibility of the administrator to identify the correct transition that needs to be forced.
- Forcing a transition will simply ignore any transition condition that might be defined for this transition in the master recipe/workflow.
- After a transition has been forced, processing continues normally, according to the SFC modeling in the underlying recipe.
- The corresponding exception is assigned to the related operation or unit procedure that contains the forced transition.
- In case a transition between unit procedures is forced, the exception is added to the preceding unit procedure that was performed prior to the forced transition. If there is no such unit procedure, the transition cannot be forced.

Archive and Purge

New for PharmaSuite 8.0.

Since PharmaSuite 8.0, the system provides the **Export for archive** function for orders and workflows.

The exported data is stored in a pre-configured target folder that cannot be changed by users. It is very important to understand that this target folder must be protected by special access rights in a similar way as the database of PharmaSuite is protected, in order to prevent unauthorized data manipulation.

Once an order or workflow has been exported for archive, it is marked as **Exported for archive** in PharmaSuite and therefore can be purged at any time.

All archive and purge-related events are tracked in a separate event log within PharmaSuite.

Restrictions for Export for Archive

Updated for PharmaSuite 8.1.

Orders can only be exported for archive if they are in the **Reviewed** or **Canceled** statuses. In case a workflow is appended to multiple orders, its status is changed to **Reviewed** along with the review of the first order to which it is assigned. Along with the export of an order, the system exports all currently appended workflows as well.

Workflows on their own can be exported for archive if they are at least in the **Finished** or **Canceled** statuses.

Exclusions from Export for Archive

New for PharmaSuite 8.0.

The following objects are not exported along with their order:

- Dispense reports, because all batch-relevant information of the Dispense report is also included in the batch report, which is part of the exported data.
- Re-print history of Dispense reports when re-printed from the Production Management Client, because it is not considered to be archive-relevant data.
- Re-print history of sublot labels when re-printed from the Production Management Client, because each re-printed label is part of the exported data. Sublot labels that were re-printed from the Production Execution Client during execution, are tracked as exceptions within the batch report.

Restrictions for Purge

Updated for PharmaSuite 8.2

Orders can be purged if they are in the **Annulled**, **Reviewed**, or **Canceled** statuses and they do no longer contain a reference to an equipment object. Orders in the **Canceled** and **Reviewed** statuses must have been successfully exported for archive first.

Workflows can be purged if they are at least in the **Finished**, **Annulled**, or **Canceled** statuses and they do no longer contain a reference to an equipment object. In case they are **Production-relevant** (per master workflow definition, since they are appended to an order, or referenced by an equipment logbook), they need to be exported for archive first (except for workflows in the **Annulled** status).

Reference to an equipment object means that there is still a binding reference to an equipment object (e.g. scale, room) and its "current order context".

Deployment and Configuration Constraints

Setup of the Version Management for Master Recipes and Master Workflows

Updated for PharmaSuite 8.4.

The configuration of the version management for master recipes and master workflows allows to define a numbering scheme for version numbers.

In order to ensure a proper system behavior, especially related to the recipe and workflow comparison function that is available since PharmaSuite 7.1, it is important that the following basic rules always apply for the numbering scheme:

- Major revision numbers (e.g. 1.0, 2.0) must never be increased if the new master recipe or master workflow version still represents the same object within the database (e.g. for status transitions from **Verification** back to **Edit** or from **Valid** to **Valid** due to an updated period of validity).
- Major revision numbers are always increased (e.g. 1.0 to 2.0) if the new master recipe or master workflow version represents a new object within the database (e.g. created with the **Save as** function).

For more information, please refer to chapter "Adapting Versioning Graphs" in Volume 2 of the "Technical Manual Configuration and Extension" [A20] (page 45).

Client Inactivity Timeout (FTPC)

Updated for PharmaSuite 8.2.

FactoryTalk ProductionCentre provides a Client Inactivity Timeout feature (defined in FTPC Administrator, see "FactoryTalk ProductionCentre Administrator Release 10.4 User's Guide" [A3] (page 45)). However, this feature is not supported by PharmaSuite. Enabling this feature may cause unforeseen timeouts.

For the PharmaSuite Production Execution Client, an automatic locking mechanism is available. In case no user activity is detected for a certain amount of time, the system locks the client with the display of the **Change User** dialog in order to prevent unauthorized activity at the Production Execution Client.

In case inactivity timeouts are desired for other PharmaSuite clients, this needs to be managed through the timeout feature of the client operating system.

Length of Configuration Key Names

The name of configuration keys must not be longer than 64 characters.

For example, your configuration key name is **LibraryHolder/services-inventory-impl.jar/AtooLongConfigurationItemName**. Then, this key cannot be stored in the database if none of the two hierarchy levels (**LibraryHolder**, **services-inventory-impl.jar**) is present in the corresponding application object, which holds the configuration key. The reason is that the complete name is longer than 64 characters.

But if at least the first hierarchy level (**LibraryHolder**) is present (which is currently the case in our **DefaultApplication** application), the key could be stored in the database. The key name would be **services-inventory-impl.jar/AtooLongConfigurationItemName**, which is less than 64 characters long.

IT Environment - SOAP not supported

Updated for PharmaSuite 7.1.

FactoryTalk ProductionCentre supports two different ways of the communication between the Application Server and the client applications:

- RMI/IIOP (default)
- SOAP/HTTP.

IMPORTANT

PharmaSuite utilizes User Transactions for consistent data management. User Transactions do not work with SOAP. Thus, RMI/IIOP is required for PharmaSuite.
Quoted from the "Process Designer Online Help": "This feature is **not** supported when using .NET proxy factory or any client using the SOAP protocol."
The system always tries to use RMI first, but will fall back to SOAP if no communication is possible based on RMI/IIOP. However, PharmaSuite does not fully support SOAP (this especially applies to User Transactions).

Running PharmaSuite with SOAP might result in a strange application behavior and potential serious data loss.

Since PharmaSuite 2.1, checks have been implemented, which apply during client and server startup, e.g., for Production Execution Client, Production Management Client, Production Response Client or Shop Operations Server startup for EBR and OE server. However, problems might occur if the communication switches to SOAP later on.

Error analysis might be difficult because Process Designer works without any problem. The same applies to the Production Execution Client until you start processing.

Therefore, the IT environment has to ensure that communication based on RMI/IIOP is continuously possible and is reliable between the Application Server and the client applications, e.g., the use of WLAN-based communication is not recommended.

IT Environment - System Time Must Be Synchronized

Make sure that the system date and time settings are synchronized between the machines that are running PharmaSuite clients and servers, including application servers, database servers, ActiveMQ JMS message brokers, and the Shop Operations Servers.

Especially in an automation or Historian integration scenario it is essential to have the system time synchronized to the second for all machines hosting the Production Execution Client application, the ActiveMQ JMS message broker, and the Automation Integration server.

Otherwise, messages sent with the expiration time of client's "now" time might be disregarded by the message broker or the Automation Integration server if they are running on a more advanced time. The message expiration time is configured at the PharmaSuite client and is set to 3 seconds for automation and Historian integration communication (see **MessageTimeToLive4AI** configuration key in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 45)). The setting prevents that outdated messages are received by an Automation Integration server which has temporarily been unavailable, e.g. in case of a network outage between message broker and Automation Integration server or an outage of the Automation Integration server itself. Particularly when writing tags via messaging it is important that outdated messages expire. We highly recommend not to increase the setting of the **MessageTimeToLive4AI** configuration key.

Please consider to configure NTP synchronization on each machine.

Since PharmaSuite 5.2, the Production Execution Client, and since PharmaSuite 6.0, all clients provide information about the system time synchronization status of the required systems in an **About** dialog.

Chapter 4

List of Known Issues

The following list of issues contains known defects that might have a business impact.

It is planned to fix these issues in a future release.

General

Disable triggers for OBJECT_COUNT in the ACTIVITY_SET table

- **ID**: FTPS-2126
- **Issue**: The OBJECT_COUNT triggers in the ACTIVITY_SET table lead to performance issues during master recipe preparation.
  Refers to PC-17094.
  This issue has been solved for PharmaSuite 5.1. However, the counter for the total number of Activity Sets is no longer correctly displayed in Process Designer.
- **Solution**: Do not rely on the Activity Set filter *get count* method without any additional filter criteria.

Timestamp handling in PS Administration

New for PharmaSuite 10.01.00

- **ID**: CVBLSFTPS-9255
- **Issue**: Timestamps are always shown in the time zone of the application server. They do not show in the local time zone of the client. PS Administration currently has two timestamp fields in the user editor: Password expiration and User expiration.
- **Solution**: Run both client and server in the same timezone.

Recipe and Workflow Designer

Units of Measure containing a division slash are not supported by the Expression editor

- **ID**: FTPS-1904
- **Issue**: The Expression editor does not provide means to support measured value constants where the measured value is used as a constant and the unit of measure contains a division slash (e.g., 100 mg/ml).
- **Solution**: Units of measure must be defined without using a "/" at all, e.g., replace "mg/ml" by "mgPml" or use the *convertToMeasuredValue()* function to mask the unit of measure with special characters.

Stack overflow error logged for recipes/workflows with a large number of phases

- **ID**: FTPS-1507
- **Issue**: When a recipe, workflow, or custom building block is built that holds more than 500 phases in one single operation, the system may respond with a stack overflow error.
- **Solution**: In general, PharmaSuite's design space is meant to support recipes, workflows, or custom building blocks with up to 1,000 phases. However, more than 500 phases within one single operation is not considered to be a realistic scenario and therefore is not supported by the system.

Data Manager - Work Center

Work Centers do not export their Production Line(s)

- **ID**: FTPS-2665
- **Issue**: Process Designer's DSX export does not export production lines attached to work centers when exporting work centers.
  Missing production lines result in orders not being available for execution in the Production Execution Client.
  Hence, on systems with imported work centers, no orders can be processed on these work centers.
  Refers to PC-17325.
- **Solution**: Export the production line containing the required work centers with Process Designer’s DSX export. Then the DSX file contains the child work centers.

Execution

Parallel operations execution: Scale communication errors

New for PharmaSuite 7.1.

- **ID**: BB-809
- **Issue**: A scale communication error occurs when parallel weighing operations are executed with two different scales but using the same station and device. This is due to the fact that the scale drivers only support single scales and only one scale driver can be loaded to a device at a time.
- **Solution**: In case a recipe definition allows to run parallel weighing operations at the same station and device, it needs to be ensured procedurally that only one weighing operation is processed at a time and that the actual weighing step is completed (scale is released), before the operator continues processing the parallel weighing operation.

Batch Report

Internationalization of Load Logistic Unit phase sub-report

New for Phase Load Logistic Unit 6.0 / 6.1

- **ID**: CVBLSFTPS-23979
- **Issue**: The phase sub-report does not show all sublots which were relocated to a logistic unit. This happens only if one of the table column headers are translated with a term that causes a wrapping of the header to a second line.
- **Solution**: If message packs with translations to further languages are available, check that no header column text causes a wrapping. If the header is wrapped to a second line, shorten the translated text.
  For newer phase versions, it is planned to have a table output that can handle column header wrapping for reports.

Environment

PharmaSuite services are not started properly after reboot of the application server

- **ID**: FTPS-61
- **Issue**: After a reboot of the Application Server, the **PharmaSuite Transition server**, **PharmaSuite EBR server**, **PharmaSuite TOM server**, and **PharmaSuite OE server** services are not restarted automatically. The reason for this is that the JBoss Service has not completed its starting procedure at that time. The JBoss server is still in the **Starting** status.
  Refers to PC-13743.

- **Solution**: Wait until the JBoss Service is in the **Started** status, then start the **PharmaSuite Transition server**, **PharmaSuite EBR server**, **PharmaSuite TOM server**, and **PharmaSuite OE server** services manually.
  For the status of the service, see *..\Rockwell\FT_ProductionCentre\jboss\server\datasweepConfig\log\server.log*.
  The line

  ```
  <timestamp> 2017-06-09 13:50:17,271 INFO  [org.jboss.as] (Controller
  Boot Thread) WFLYSRV0025: JBoss EAP 7.0.6.GA (WildFly Core
  2.1.8.Final-redhat-1) started in 14837ms - Started 1782 of 2126
  services (518 services are lazy, passive or on-demand)
  ```

  indicates that the service is started. This is the last line in the log file after the service was started.
  Please note that the version numbers mentioned in the log file may be higher than the one shown here.

Upgrade-related Topics

The following list of topics refers to issues that are directly related to system updates or data migration of PharmaSuite.

Upgrade Engine-related Issues

Migration Execution Can Only Be Canceled Between Stages

Within a running stage of the migration execution (e.g. stage 3 with multiple update tasks), the execution cannot be canceled. The system first completes all pending tasks of this stage, in order to keep data and log files consistent. The migration execution can only be canceled upon completion of the actual stage.

Migration of Audit Trail Configuration Settings Is a Manual Step

New for PharmaSuite 8.0.

While upgrading to a new PharmaSuite release, a manual configuration is required in order to disable audit trail of specific database tables.
For more information, please refer to section "Updating the Audit Trail Configuration", chapter "Upgrading an Installation" in "Technical Manual Installation - Upgrade" [A8] (page 45).

Transaction Timeout JBoss

Updated for PharmaSuite 10.0.

Due to a changed JBoss behavior since JBoss 7.2.6, the transaction timeout can no longer be configured with JBoss configuration. A default user transaction timeout is defined by the **TransactionTimeout** and **SuspendedTransactionTimeout** configuration keys.

For more information, please refer to chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 45).

Specifics Related to PharmaSuite 10.02.00

Discontinuation of Specific EBR Phase Building Blocks

The **Get text value**, **Get process value**, and **Get choice value** phase building blocks will be discontinued in 2021. They are still supported, however new features will no longer be integrated. Starting with PharmaSuite 9.2, we recommend to start replacing these phase building blocks by the **Get values** phase building block.

The **Show document** phase building block will be discontinued in 2021.

Discontinuation of Specific DCS Phase Building Block

The **Get DCS batch values** phase building block will be discontinued in 2021. It is still supported, however new features will no longer be integrated. Starting with PharmaSuite 9.1, we recommend to start replacing the phase building block by the **Get DCS parameters** phase building block.

User Audit Trail

The user audit trail shall track only changes on a user's master data e.g., description or email address. Dynamic runtime data (user login, user logout, and failed login (additionally depends on used security provider)) is not anymore available in a user's audit trail data. Please refer to the system access history if this information is required.

Exception Status and None Risk

PharmaSuite behavior prior to 10.02.00: Exceptions with the **None** risk are created in the **Open** status and their status can be changed to any other status. But it is possible to keep these exceptions in the **Open** status even if the related order or workflow is already in the **Reviewed** status.

PharmaSuite behavior since 10.02.00: Exceptions with the **None** risk always have the **N/A** status. Comments can be added, but the status cannot be changed. After changing the risk to a different value, the status switches automatically to **Open** and can be changed to any value except **N/A**. If the risk of an exception is changed to **None**, the status is switched automatically to **N/A**, independent of the current status.

Exception in the Closed Status

PharmaSuite behavior prior to 10.02.00: Any user who has the access privilege to add a comment to any exception can do this as well for an exception in the **Closed** status (as long as the order or workflow is not **Reviewed**).

PharmaSuite behavior since 10.02.00: As changing a **Closed** exception is not seen as common task for a reviewer, now to add a comment to an exception in the **Closed** status, an additional access privilege is required for the user, in the Production Execution Client PEC_AccessRight_AddCommentForClosedException and in the Production Response Client PRC_AccessRight_AddCommentForClosedException.

System Building Block Compatibility

Due to the nature of some code changes in the latest PharmaSuite release, not all product building block versions of the previous release are compatible with the latest release but need to be updated to their newest building block maintenance release version first.

For more details about the specific building block versions that are compatible with the latest PharmaSuite release, please refer to "PharmaSuite Building Blocks - Compatibility Matrix" [A13] (page 45).

System Building Block Replacement through Change Request

Since PharmaSuite 10.0, system building blocks can be replaced by another version using the change request feature. In many cases, new or changed functionality is introduced with a new system building block version that needs additional configuration. The recipe author needs to do this configuration before the target object is approved. So, the target status defined at the change request for the recipe/workflow or building block to create shall be a Draft status.

The following table provides an overview of all new minor or major system building block versions delivered with PharmaSuite 10.02.00. For details, please refer to the PharmaSuite Functional Requirement Specification of the system building block.

System Building Block old Version to new Version	No change or New / Changed parameter, default is previous behavior	New / Changed parameter, needs configuration or has changed default behavior	New exception parameter, optional to change risk / text	Remark
Get DCS Alarms 1.0 to 1.1	Yes	No	No	New process parameters: Separated exception per converted or unconverted alarm. It is now configurable if all at once converted alarms are added as separated comments to only one exception, or all at once converted alarms are created as individual exceptions, one per alarm. The same applies to unconverted alarms. It is configurable if one exception is created per unconverted alarm, or only one exception is created, and all unconverted alarms are added as separated comments.

Specifics Related to Older PharmaSuite Releases

Please refer to the Release Notes of the specific PharmaSuite release.

Reference Documents

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

No.	Document Title	Part Number
A1	PharmaSuite Glossary	PSGY-QR009C-EN-E
A2	PharmaSuite Technical Manual Installation	PSES-IN010C-EN-E
A3	FactoryTalk ProductionCentre Administrator Release 10.4 User's Guide Set Client Inactivity Timeout	PCADM-IN104A-EN-E
A4	PharmaSuite Technical Manual Installation - Building Blocks	PSBB-IN009C-EN-E
A5	PharmaSuite Technical Manual Administration	PSAD-RM010C-EN-E
A6	PharmaSuite Technical Manual Configuration & Extension - Volume 4	PSCEV4-GR010C-EN-E
A7	PharmaSuite Technical Manual Configuration & Extension - Volume 3	PSCEV3-GR010C-EN-E
A8	PharmaSuite Technical Manual Installation - Upgrade	PSUP-IN007C-EN-E
A9	PharmaSuite Technical Manual Developing System Building Blocks	PSBB-PM010B-EN-E
A12	PharmaSuite Functional Requirement Specification Dispense and Inline Weighing	PSFRSDI-RM008B-EN-E
A13	PharmaSuite Building Blocks - Compatibility Matrix	PSBBCM-PA010C-EN-E
A14	PharmaSuite Implementation Guide PS Administration	PSAC-IN002C-EN-E
A15	PharmaSuite Functional Requirement Specification Data Management	PSFRSDM-RM006C-EN-E
A16	PharmaSuite Functional Requirement Specification Output Weighing	PSFRSOW-RM004B-EN-E
A17	PharmaSuite Functional Requirement Specification Recipe and Workflow Management	PSFRSRD-RM010C-EN-E
A18	FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced	PCJBAD IN104A EN E
A19	FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Stand-Alone	PCJBSA IN104A EN E
A20	PharmaSuite Technical Manual Configuration & Extension - Volume 2	PSCEV2-GR010C-EN-E
A21	FactoryTalk ProductionCentre 10.4 Supported Platforms Guide	PRDCTR-RM104A-EN-E
A22	PharmaSuite Supported Platforms Guide	PSPG-RM010C-EN-E

No.	Document Title	Part Number
A23	PharmaSuite Functional Requirement Specification Execution Framework	PSFRSEF-RM006C-EN-E
A24	PharmaSuite Technical Manual Configuration & Extension - Volume 5	PSCEV5-GR007B-EN-E

TIP

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

The following documents are available upon request.

No.	Document Title	Part Number
A10	Technical Description - Number Handling	10005511806/SPC
A11	Technical Description - Quantity Calculations	10005444208/SPC

Appendix A - Document Information

The document information covers various data related to the document.

Approval

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

Name	Role
Andreas Grossmann	Product Manager
Norbert Ern	Product Owner
Steffen Landes	Engineering Manager
Jürgen Stieber	Technical Lead
Eva Müller	Test Manager
Ignaz Wangler	Test Lead

Revision History

The following table describes the history of this document.

Version	Description
1.0	Release-specific Enhancements and Changes: updated (page 3) Limitations: updated (page 4)
1.0	Information and Tips ■ Added: Equipment Automation Tag Changes (page 15) ■ Removed: Export and Import of Master Recipes and Master Workflows
1.0	List of Known Issues ■ Added: Batch Report - Internationalization of Load Logistic Unit phase sub-report (page 39)
1.0	Upgrade-related Topics ■ Added: Specifics Related to PharmaSuite 10.02.00 (page 42) ■ Removed: Specifics Related to PharmaSuite 10.01.00
1.0	Technical Release Notes ■ Added: Technical Release Notes - PharmaSuite 10.02.00 (page 49) ■ Removed: Technical Release Notes - PharmaSuite 10.01.00
1.0	Open Source License Agreements (page 53) ■ Updated: Apache Commons DNCP 2.8.0 ■ Updated: Apache Commons Net 3.8 ■ Updated: Apache Commons Pool 2.9.0 ■ Updated: Apache Commons Text 1.9 ■ Updated: Apache PDFBox 2.0.24 ■ Updated: ZXing 3.4.1

Appendix B - Technical Release Notes

Technical Changes of the Underlying Platform

This section lists technical changes of the underlying platform which are of interest for system integrators who are going to use PharmaSuite based on FactoryTalk ProductionCentre 10.4 MR7 (10.4.107937).

- No changes to previous release

Technical Release Notes - PharmaSuite 10.02.00

In PharmaSuite 10.02.00, existing components have been changed, which is of interest for system integrators who have already implemented features based on the components of the published API.

Modified component: New base class for process parameters

- **ID**: CVBLSFTPS-17180
- **Before**: When using the Parameter Class generator to create a new parameter class, the following two classes are created (among others):
  - *MESGeneratedParam<ParamName>* extends *BulkSaveableMESATObject*
  - *MESParam<ParamName>* extends *MESGeneratedParam<ParamName>* implements *IMESProcessParameterData*
- **After**: A new abstract class is introduced: *AbstractProcessParameterData*. It extends *MESATObjectSupportingBulkDBOperations* and implements *IMESProcessParameterData*.
  When using the Parameter Class generator to create a new parameter class, the definitions of the two classes now look like this:
  - *MESGeneratedParam<ParamName>* extends *AbstractProcessParameterData*
  - *MESParam<ParamName>* extends *MESGeneratedParam<ParamName>*
- **Reason**: It is recommended to use *MESATObjectSupportingBulkDBOperations* rather than *BulkSaveableMESATObject* as it also supports bulk delete. It was decided to introduce an additional base class for process parameters, which extends this class.

- **Migration**: You do not need to migrate your existing parameter classes, just keep in mind that future parameter classes will have a different class hierarchy.

Modified component: New service for export and import of recipe structures

- **ID**: CVBLSFTPS-16635
- **Before**: In the *com.rockwell.mes.apps.recipeb2mml* package there was a (non-public) API for exporting and importing recipe structures to and from BML files, respectively. This API was not public anyway, however, some customer projects may have leveraged it to realize a project-specific export/import functionality.
- **After**: Now there is a new public service *IRecipeStructureExportImportService* located in the *com.rockwell.mes.apps.recipeeditor.ifc.importexport* package, which should be used to implement project-specific export/import functionality.
- **Reason**: It was decided to not deliver the existing stand-alone export and import forms anymore but instead, to relocate the export and import of recipe structures completely to Recipe and Workflow Designer. Therefore, the existing API located in the *com.rockwell.mes.apps.recipeb2mml package* is no longer supported; it is no longer stable, was massively adapted, and may be removed completely in the future anyway.
- **Migration**: Use the new service in case you have used the (non-public) API in *com.rockwell.mes.apps.recipeb2mml* before.

Modified component: IDataBuilder

- **ID**: CVBLSFTPS-16671
- **Before**: The *com.rockwell.mes.services.batchrecord.ifc.IDataBuilder* interface extended *IMESService*.
- **After**: The *IDataBuilder* interface no longer extends *IMESService*.
- **Reason**: Data builders need to be stateful to support persisted messages from import.
- **Migration**: When creating an instance of a builder replace *ServiceFactory.getService(Class<IServiceInterface>)* with *ObjectFactory.createObject(Class<IObjectFacade>)*.

Modified component: LDVerifyTagsResult

- **ID**: CVBLSFTPS-17208
- **Before**: *com.rockwell.mes.commons.livedata.ifc.LDVerifyTagsResult.setWrongTags(List<String>)*
- **After**: Removed.

- **Reason**: A lot of boilerplate code can be avoided if string lists are fully handled inside the *LDVerifyTagsResult* instance.
- **Migration**: Use *com.rockwell.mes.commons.livedata.ifc.LDVerifyTagsResult.addWrongTag(String)* to add a single string to the *LDVerifyTagsResult* instance.

Modified component: LDVerifyTagsResult

- **ID**: CVBLSFTPS-17208
- **Before**: *com.rockwell.mes.commons.livedata.ifc.LDVerifyTagsResult.setWrongTypeTags(List<String>)*
- **After**: Removed.
- **Reason**: A lot of boilerplate code can be avoided, if string lists are fully handled inside the *LDVerifyTagsResult* instance.
- **Migration**: Use *com.rockwell.mes.commons.livedata.ifc.LDVerifyTagsResult.addWrongTypeTag(String)* to add a single string to the *LDVerifyTagsResult* instance.

Modified component: LDVerifyTagsResult

- **ID**: CVBLSFTPS-17208
- **Before**: *com.rockwell.mes.commons.livedata.ifc.LDVerifyTagsResult.setWrongTagsType(List<String>)*
- **After**: Removed.
- **Reason**: A lot of boilerplate code can be avoided, if string lists are fully handled inside the *LDVerifyTagsResult* instance.
- **Migration**: Use *com.rockwell.mes.commons.livedata.ifc.LDVerifyTagsResult.addWrongTagType(String)* to add a single string to the *LDVerifyTagsResult* instance.

Modified component: LDVerifyTagsResult

- **ID**: CVBLSFTPS-17208
- **Before**: *com.rockwell.mes.commons.livedata.ifc.ILiveDataConnector.verifyTagsOfLiveDataGroup(String, LDGroupDetails)*
- **After**: *com.rockwell.mes.commons.livedata.ifc.ILiveDataConnector.verifyTagsOfLiveDataGroup(LiveDataGroupDescriptor, LDGroupDetails)*

- **Reason**: The new method uses the *LiveDataGroupDescriptor* to recreate a read/write *LiveDataGroup* for all automation properties of an equipment entity if the verification succeeds and if the currently available live data group differs from the expected one, provided by the *LiveDataGroupDescriptor*.
- **Migration**: Use *com.rockwell.mes.commons.livedata.ifc.ILiveDataConnector.verifyTagsOfLiveDataGroup(LiveDataGroupDescriptor, LDGroupDetails)*.

Modified component: IAutomationService

- **ID**: CVBLSFTPS-17208
- **Before**: *List<String>[] com.rockwell.mes.services.s88equipment.ifc.automation.IAutomationService. verifyTagsAndTypesOfLiveDataGroup(String, String, LDGroupDetails)*
- **After**:
  - *TagVerificationResult com.rockwell.mes.services.s88equipment.ifc.automation.IAutomationService. verifyTagsAndTypesOfLiveDataGroup(String, String, LDGroupDetails)*
  - *TagVerificationResult com.rockwell.mes.services.s88equipment.ifc.automation.IAutomationService. verifyTagsAndTypesOfLiveDataGroup(IMESS88Equipment, LDGroupDetails)*
- **Reason**: A lot of boilerplate code can be avoided, if string lists are fully handled inside the *TagVerificationResult* instance.
- **Migration**: Extract the lists of interest from *TagVerificationResult*. Also prefer the new method, which passes an *IMESS88Equipment* as argument and also keeps the read/write group of the equipment entity up to date. If you use the old version, you have to maintain the LiveData groups yourself.

Appendix C - Open Source License Agreements

PharmaSuite 10.02.00 includes Open Source software as indicated in the Release Notes of FactoryTalk ProductionCentre. The Release Notes are provided along with the FactoryTalk ProductionCentre installation.

Furthermore, PharmaSuite 10.02.00 includes specific Open Source software as listed below. For details of the license agreements, follow the links.

- Apache Commons Configuration 2.7
  Apache License Version 2.0, January 2004 (page 54)
- Apache Commons DBCP 2.8.0
  Apache License Version 2.0, January 2004 (page 54)
- Apache Commons Net 3.8
  Apache License Version 2.0, January 2004 (page 54)
- Apache Commons Pool 2.9.0
  Apache License Version 2.0, January 2004 (page 54)
- Apache Commons Text 1.9
  Apache License Version 2.0, January 2004 (page 54)
- Apache PDFBox 2.0.24
  Apache License Version 2.0, January 2004 (page 54)
- Apache JBig2-Imageio 3.0.3
  Apache License Version 2.0, January 2004 (page 54)
- JGraph 5.12.4.0
  GNU Lesser General Public License Version 2.1, February 1999 (page 56)
- jSerialComm 2.6.2
  Apache License Version 2.0, January 2004 (page 54)
- FreeMarker 2.3.15
  FreeMarker 2.3.15 License, 2003 (page 62)
- FMPP 0.9.14
  FMPP 0.9.14 License, 2003 (page 62)
- ANTLR 3.5.2
  BSD License, 2010 (page 63)
- izPack 5.0
  Apache License Version 2.0, January 2004 (page 54)

- ZXing 3.4.1
  Apache License Version 2.0, January 2004 (page 54)

Apache License Version 2.0, January 2004

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work,

where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
(b) You must cause any modified files to carry prominent notices stating that You changed the files; and
(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.
You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions.
Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]
Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0
Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and limitations under the License.

GNU Lesser General Public License Version 2.1, February 1999

GNU Lesser General Public License
GNU LESSER GENERAL PUBLIC LICENSE
Version 2.1, February 1999
Copyright (C) 1991, 1999 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

Preamble
The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.
This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.
When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.
To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.
For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all

the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

GNU LESSER GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

----

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version

than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.)   Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.
This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.
If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library".   Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.
However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library".   The executable is therefore covered by this License.

Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not.
Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library.   The threshold for this to be true is not precisely defined by law.
If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work.   (Executables containing this object code plus portions of the Library will still fall under Section 6.)
Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6.
Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

----

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.
You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License.   You must supply a copy of this License.   If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License.   Also, you must do one of these things:
a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library.   (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.
c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.
d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.
e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.
For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.
It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

----

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:
a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.
b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

----

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain

entirely from distribution of the Library.
If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.
It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.
This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.
Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

----

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

© 2001-2009 - JGraph - FAQ - Contact - About

FreeMarker 2.3.15 License, 2003

FreeMarker 1.x was released under the LGPL license. Later, by community consensus, we have switched over to a BSD-style license. As of FreeMarker 2.2pre1, the original author, Benjamin Geer, has relinquished the copyright in behalf of Visigoth Software Society. The current copyright holder is the Visigoth Software Society.

Copyright (c) 2003 The Visigoth Software Society. All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. The end-user documentation included with the redistribution, if any, must include the following acknowlegement:
"This product includes software developed by the Visigoth Software Society (http://www.visigoths.org/)."
Alternately, this acknowlegement may appear in the software itself, if and wherever such third-party acknowlegements normally appear.

3. Neither the name "FreeMarker", "Visigoth", nor any of the names of the project contributors may be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact visigoths@visigoths.org.

4. Products derived from this software may not be called "FreeMarker" or "Visigoth" nor may "FreeMarker" or "Visigoth" appear in their names without prior written permission of the Visigoth Software Society.

THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE VISIGOTH SOFTWARE SOCIETY OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Visigoth Software Society. For more information on the Visigoth Software Society, please see http://www.visigoths.org/

FMPP 0.9.14 License, 2003

Copyright (c) 2003, Dániel Dékány
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

* Neither the name "FMPP" nor the names of the project contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

BSD License, 2010

[The BSD License]
Copyright (c) 2010 Terence Parr
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the author nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Index