LISTEN.
THINK.
SOLVE.®

PharmaSuite®

CONFIGURATION AND EXTENSION – VOLUME 4

RELEASE 8.4

TECHNICAL MANUAL

PUBLICATION PSCEV4-GR008E-EN-E-DECEMBER-2017
Supersedes publication PSCEV4-GR008D-EN-E

Allen-Bradley • Rockwell Software

Rockwell Automation

Contact Rockwell See contact information provided in your maintenance contract.

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

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

Contents

Figures

Chapter 1

Introduction

This documentation contains important information about how to configure and adapt a PharmaSuite system including information about actions that need to be performed in FactoryTalk® ProductionCentre.

The information is structured in the following sections:

Volume 1: Style and Layout

In this volume you will find information about standard styles and layouts that are defined for the user interface of PharmaSuite. You will also learn how to modify the elements of the user interface to fit your purposes.
It consists of the following chapters:

- Changing the General Appearance of PharmaSuite
- Creating and Adapting Forms
- Using Forms in the Production Execution Client
- Using Forms in the Production Management Client
- Adapting the Workflows of the Production Execution Client
- Adapting the Use Cases of the Production Management Client
- Changing the Layout of Forms

Volume 2: Semantic (affecting the whole system)

In this volume you will find information about such system functionalities as field attributes, version management, services, audit trail, etc. You will also learn how you can adapt the mechanisms available in PharmaSuite to match your needs, as well as maintain and manage certain elements of the system, such as units of measures, signatures, or users.
It consists of the following chapters:

- Adapting and Adding Field Attributes
- Creating and Extending GUI Activities
- Creating Java Artifacts to Access Application Tables
- Configuring Flexible State Models
- Adapting Versioning Graphs
- Localizing Date and Time Representation

- Managing Services
- Adapting the Order Explosion Service
- Documenting the Change History of Order Definitions
- Managing Electronic Signatures and Access Rights
- Modifying and Adding Constraints
- Modifying and Adding Checks
- Managing Audit Trail
- Adding Units of Measure and Global Conversion Definitions
- Changing Number Generation Schemes
- Changing or Adding New Reports
- Changing or Adding New Labels
- Working with the Batch Record API
- Adapting the Export for Archive Process
- Adapting the Purge Process
- Defining the Risk Level for Exceptions
- Supporting External Authentication Systems
- Supporting External Exception Review
- Defining Units of Measure for Potency
- Adapting the About Dialog
- Adding New Scale Drivers
- Providing Localization to Support Different Locales

VOLUME 3: SEMANTIC (AFFECTING SPECIFIC AREAS)

In this volume you will find information about providing additional information in Recipe and Workflow Designer or Production Execution Client, etc. You will also learn how you can adapt windows in Recipe and Workflow Designer, Cockpit actions, or user documentation.
It consists of the following chapters:

- Adding Numeric Fields to the Database
- Configuring the Universe of Recipe and Workflow Designer
- Configuring the Setlist of Recipe and Workflow Designer
- Configuring the Property Windows of Recipe and Workflow Designer
- Configuring the Parameter Panel of Recipe and Workflow Designer

- Managing the Layout of Recipe and Workflow Designer and Data Manager
- Configuring the Expression Editor of Recipe and Workflow Designer and Data Manager
- Using ERP BOMs in Recipe and Workflow Designer
- Configuring the Initialization of Material Parameters in Recipe and Workflow Designer
- Configuring Menu and Toolbar of Recipe and Workflow Designer
- Configuring the Details Window of Data Manager - Work Center
- Configuring the Change History of Data Manager - Work Center
- Configuring FSMs for Equipment Properties
- Providing Images for Equipment Classes in Data Manager
- Providing Report Designs for (Template) Equipment Entities in Data Manager
- Adding an Equipment Logbook Category Accessible by Phase Building Blocks
- Retrieving Specific Information from the Equipment Logbook
- Configuring Additional Purposes for the Equipment Type Technical Property Type
- Adding a Workflow Type to Workflow Designer and the Production Execution Client
- Managing Cockpit Actions of the Production Execution Client
- Configuring Menu and Toolbar of the Production Response Client
- Adding Filter Attributes to the Exception Dashboard of the Production Response Client
- Material Handling for DCS
- Adapting User Documentation

VOLUME 4: CONFIGURATION FRAMEWORK

In this volume you will find information about the methods for managing application configurations, logging and debugging. You will also learn how to administer configuration keys, and which configuration keys are available in PharmaSuite.
It consists of the following chapters:

- Managing Configurations (page 11)
- Logging and Debugging (page 21)
- Configuration Keys of PharmaSuite (page 29)

VOLUME 5: EXTENSION USE CASES

In this volume you will find step-by-step instructions for selected extension use cases. As a trained FactoryTalk ProductionCentre and PharmaSuite system integrator, you will learn how to adapt PharmaSuite.
It consists of the following chapters:

- Processing Materials with a Second Potency
- Adding the Second Potency Attribute to ERP BOMs
- Adding the Second Potency Function to the Expression Editor
- Managing Batches in the ConditionallyReleased Status
- Adapting the Approval Workflow of Master Recipes
- Adding the Material Balance Section to the Batch Report
- Displaying a Retest Date Column in the Overview Panel of the Exception Dashboard
- Adding the Print FDA Report Function to PharmaSuite Clients
- Adding the Print QA Report Function to Recipe and Workflow Designer
- Automatic Archiving and Purging of Specific Orders and Workflows
- Displaying Custom Columns in PharmaSuite's Audit Trail
- Using a Locale with Unicode Fonts

REFERENCE DOCUMENTS

Finally, the Reference Documents (page 141) section provides a list of all the documentation that is referenced in this manual.
In Volume 5, each extension use case description comes with its own Reference Documents sections.

Intended Audience

This manual is intended for system engineers who implement customer-specific configurations and extensions to a PharmaSuite standard system.

The system engineers need to have a thorough working knowledge of FactoryTalk ProductionCentre, PharmaSuite, and other components relevant to the configuration or extension use cases.
Some use case require a fully set up PharmaSuite development environment and profound Java know-how.

Typographical Conventions

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

Bold typeface — Designates user interface texts, such as

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

Italic typeface — Designates technical background information, such as

- path, folder, and file names
- methods
- classes.

CAPITALS — Designate keyboard-related information, such as

- key names
- keyboard shortcuts.

`Monospaced typeface` — Designates code examples.

Chapter 2

Extension and Naming Conventions

This section describes how to control efforts for migrating your PharmaSuite installation to upcoming versions. If you cannot observe the guidelines for technical reasons, please report the issue to your dedicated delivery team of Rockwell Automation or your system integrator.

PharmaSuite artifacts fall into two main groups: building blocks and PharmaSuite core artifacts.

- When you wish to modify a **building block**, use a copy of the building block. For details, see "Technical Manual Developing System Building Blocks" [A1] (page 141).
- When you wish to modify a **PharmaSuite core artifact**, the extension strategy depends on the artifact itself.
  Modify **DSX objects** and copy **all other PharmaSuite artifacts** (e.g. XML configurations for services).

In all cases, please follow the guidelines to control the migration effort (page 7).

Guidelines to Control the Migration Effort

Please ensure that you observe the guidelines listed below.

1. Retain the published API
   The published API of PharmaSuite is accessible via the PharmaSuite start page ("PharmaSuite-related Java Documentation" [C1] (page 142)).
   - If you adapt PharmaSuite on Java level, you must only use the published PharmaSuite Java API.
     Published interfaces will only be changed if necessary. Changes to these interfaces and classes will be announced in future release notes.
   - Avoid using methods and classes that are not published, classes from the implementation package (...*impl*...), or Pnuts functions from any PharmaSuite subroutine.
     These classes and functions may change in future versions of PharmaSuite without notification.

2. Rather modify than copy
   Whenever you wish to extend/modify any object of PharmaSuite in Process Designer (DSX objects), just override the object, except for the following objects:

   - For the **Application** object **Default Configuration**, apply the mechanism of nested configurations in order to reduce a potential migration effort (see chapter "Managing Configurations" (page 11) in Volume 4 of the "Technical Manual Configuration and Extension").
   - For **FSMs** (flexible state models), structural changes (i.e. changes to states and transitions) are not allowed in order to enable a later system migration. Modifications of semantic properties can be applied to the standard FSM. They do not impact a system migration.

   When you migrate your PharmaSuite installation to another version, PharmaSuite Update and Migration displays a warning if a standard object was changed. Subsequently, you can decide whether to adapt your extension, ignore the changes delivered with the new version, or replace your extension with the new version (if applicable).

3. Mark your objects
   When naming your artifacts (e.g. objects in Process Designer, classes, interfaces, methods, functions, building blocks), use specific prefixes for your objects. The main purpose of the naming conventions is to prevent naming conflicts with deliverables from other vendors or with other versions.

   - Define and make use of a vendor code consisting of up to three uppercase letters as prefix (e.g. MYC for the My Company vendor code).
     The **X_** and **RS_** prefixes are reserved for PharmaSuite and PharmaSuite-specific product building blocks, respectively.
   - Do not reuse any of the prefixes of PharmaSuite objects in Process Designer in order to avoid conflicts during migration.
   - This guideline also applies to UDA definitions and column names of application tables.
   - Additional conventions apply to building blocks (page 9).

If you do not observe the guidelines, an update process during system migration may fail due to conflicts.

Building Block-specific Conventions

Besides the general conventions (page 7), additional conventions apply to building blocks related to vendor code, version number, and length restrictions:

1. Vendor code
   - It must be appended to the name of a phase or parameter class used in the UI (enclosed in round brackets).
   - It must be used as prefix for the AT definitions.
   - The package name must also contain a vendor reference. You can either use the vendor code or write out the company's full name.
2. Version number
   - The version number consists of two components, an integral part to refer to a major version and a fractional part to refer to a minor version, e.g. 2.1.
   - It must be appended to the name of the phase or parameter class used in the UI and to the base name used for the generated artifacts.
   - For the UI, the version number is enclosed in square brackets, e.g. [2.1].
   - Internal names must not contain brackets and dots, since Java does not allow the usage of these characters. Therefore, the last four characters are reserved for the version number, with digits 1 and 2 representing the major version and digits 3 and 4 representing the minor version. The format is xxyy, e.g. 0201 for version [2.1], 0100 for version [1.0], or 0113 for version [1.13].
3. Length restrictions
   - The maximum length of the **name** of a phase or parameter class used in the UI is 64 characters.
   - The maximum length of the **base name** of a phase building block or parameter class is 18 characters (14 for the name, 4 for the version).

Examples:

- **Hello World** phase of **My Company** with vendor code **MYC** in version **2.1**

```
<Name>Hello World Phase (MYC) [2.1]</Name>
<PhaseLibBaseName>HelloWorld0201</PhaseLibBaseName>
<ATDefinitionPrefix>MYC</ATDefinitionPrefix>
<PackageName>com.mycompany.phase.helloworld</PackageName>
```

- **My Parameter** parameter class of **My Company** with vendor code **MYC** in version **1.0**

```
<Name>My Parameter (MYC) [1.0]</Name>
<ParamClassBaseName>MyParam0100</ParamClassBaseName>
<ATDefinitionPrefix>MYC</ATDefinitionPrefix>
<PackageName>com.mycompany.parameter.myparam</PackageName>
```

Oracle Database Data Types: varchar2 and nvarchar2

When you define text fields for FactoryTalk ProductionCentre application tables or UDAs, you should be aware that there are significant differences between the **varchar2** and **nvarchar2** data types used by Oracle databases.

- The maximum field length for both database data types is restricted to 4000 bytes.
- A field of the **nvarchar2** data type works as expected: For a 4000 characters text field, you can only insert the maximum of 2000 2-byte UTF8 characters or 1300 3-byte UTF8 characters. PharmaSuite text input fields check the number of bytes.
- For an Oracle 11 database, a field of the **varchar2** data type can only handle 1000 2-byte UTF8 characters or 667 3-byte UTF8 characters. In this case, PharmaSuite text input fields check the byte length and prevent the database exception "ORA-01704: string literal too long".

TIP

Previous versions of FactoryTalk ProductionCentre (prior to 9.3) and PharmaSuite (prior to 5.0) contained **varchar2** database data type definitions instead of **nvarchar2**. Therefore a migrated PharmaSuite system may contain **varchar2** definitions and the maximum length of a **varchar2** database field may be defined in maximum number of **bytes**. In this case, if a text does not only contain 1-byte UTF8 characters, the **byte** length is greater than the **char** length and can cause a "value too large exception" (ORA-01401: inserted value too large for column, ORA-12899: value too large for column).
For this reason, we recommend to migrate all **varchar2** fields to **nvarchar2** fields. A migrated FactoryTalk ProductionCentre database should have the same database schema as the database of a newly installed FactoryTalk ProductionCentre system and only contain **nvarchar2** database data type field definitions.

Chapter 3

Managing Configurations

This section contains general information about application configurations (page 11), hierarchical configurations in PharmaSuite (page 11), and the dynamic change of a configuration (page 19). It also provides the characteristics of configuration keys (page 14), how to define configuration keys (page 20), and how to override configuration keys in unit tests (page 20).

To define a configuration key, perform the following steps:

1. Define (page 20) the configuration key.
2. Run the application with the updated configuration key settings.
3. Verify the system behavior.

What Are Application Configurations?

Application configurations are a prerequisite for the administration of configuration keys. Configuration keys are defined in the FactoryTalk ProductionCentre **Application** objects.

For information on application objects and how to change configuration keys, please refer to the FactoryTalk ProductionCentre documentation. In particular, see sections "Applications" and "CONFIGURATION_PROPERTY" in "Process Designer Online Help" [B1] (page 141).

What Are Hierarchical Configurations?

PharmaSuite supports multi-level hierarchical configurations in order to provide a flexible support for adapting and migrating multiple applications.

Typical use cases for hierarchical configurations are:

- separation of standard, global customer, and customer-site specific configuration keys
- overwriting global configuration keys for specific stations, user groups, or users.

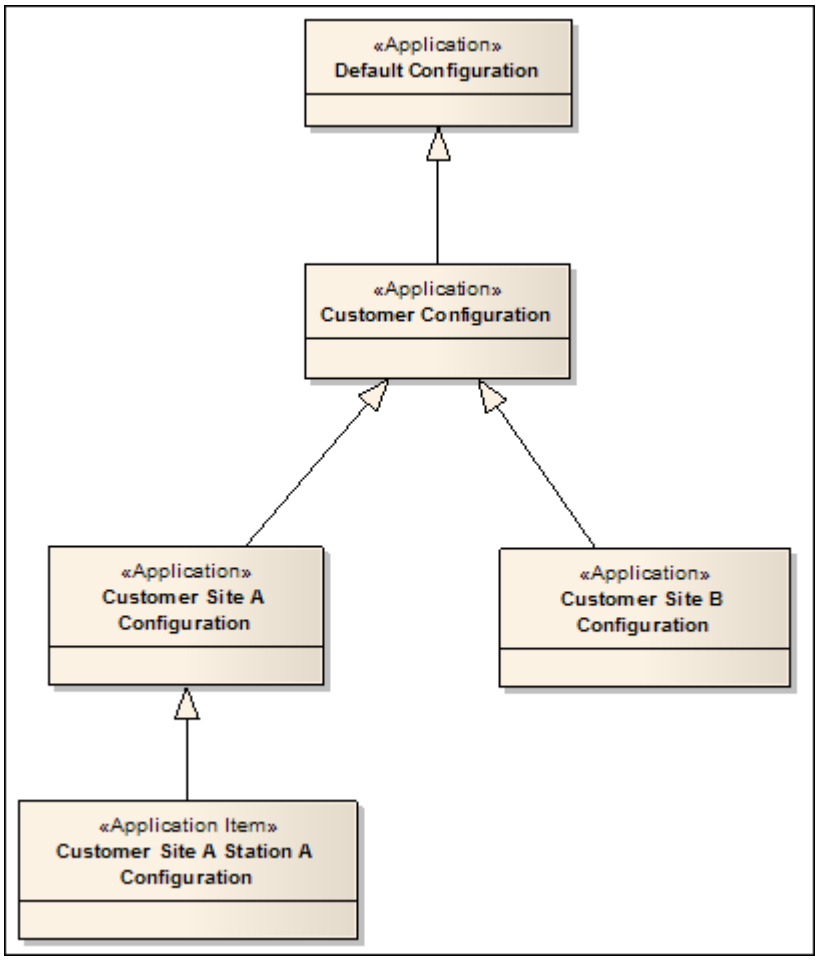

Figure 1: Example of the separation of configuration items using nested application objects

A **hierarchical configuration** is defined by linking multiple **Application** objects in a tree. The top-level root of the tree is represented by the **Application** object **Default Configuration**, which must not be modified.

The **configuration keys** defined on the **Default Configuration** level are overwritten by customer-specific configuration keys, which are defined on the **Customer** level (here: **Customer Configuration**). The customer configuration keys are overwritten by items defined on the **Customer Site** level (here: **Customer Site A Configuration**, **Customer Site B Configuration**), etc.

The **most specific *Application* object** is defined by the **BootstrapApp** parameter of a **User**, **User group**, or **Station** object (via the **Customizer** dialog). At runtime, this parameter is first searched in the current **User** object, then in its **User group** objects, and finally in the current **Station** object.

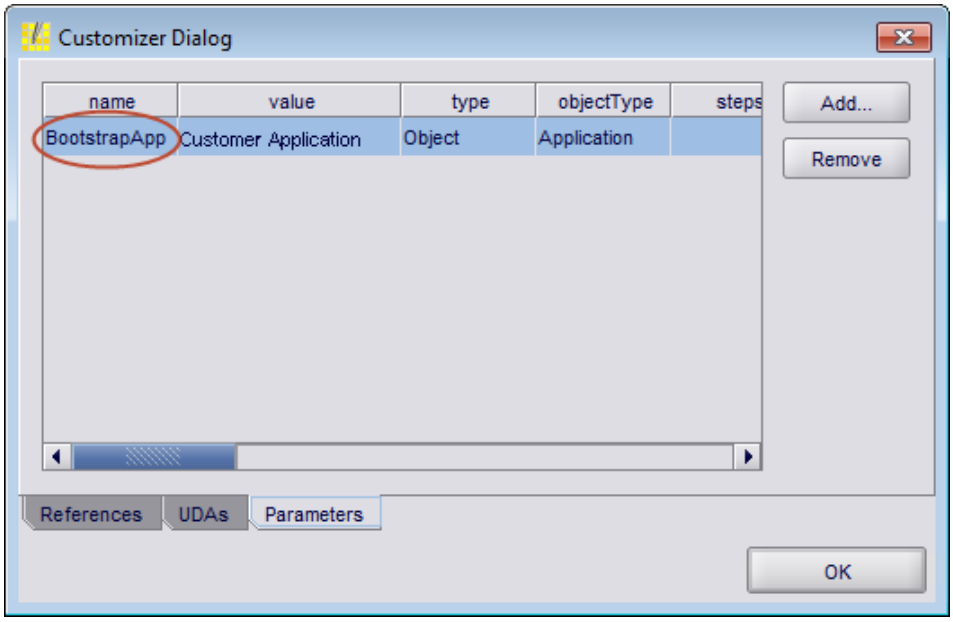

Figure 2: Define a bootstrap application

For the definition of an **application configuration hierarchy**, the **Bootstrap Application** object references another **Application** object, which also refers to another **Application** object, and so on. Here, the string of references ends with the **Application** object **Default Configuration**. The hierarchy is built up in the details frame of the **Application** object by dragging and dropping a nested **Application** object (here: **Default Configuration**) into a more specific **Application** object (here: **Customer Application**).

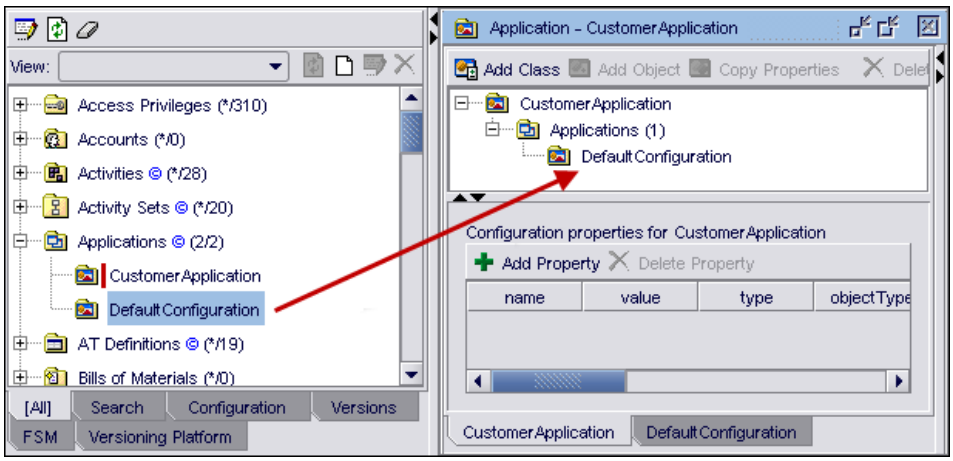

Figure 3: Build up an application hierarchy tree

This approach allows you to overwrite configuration keys by adding them to a more specific **Application** object in the configuration tree.

TIP

We do not recommend to change configuration keys directly on the top-level application **Default Configuration**.

Version Information of PharmaSuite JAR Files

The PharmaSuite JAR files are maintained as **Library** objects in Process Designer. The JAR file's version is included in the JAR file name and the **name** and **userVersion** attributes of the **Library** object.

Characteristics of Configuration Keys

A configuration key consists of three components:

[<type-name>[/<instance-name>]/]<property-key>

- <type-name>
  FactoryTalk ProductionCentre buildtime object class. The buildtime object must be supported (page 17).
  Optional component.
- <instance-name>
  Particular FactoryTalk ProductionCentre buildtime object instance of the type defined by **<type-name>**.
  Optional component, requires **<type-name>**.
- <property-key>
  Key itself. It must not contain any slashes ("/"). If you need to create a grouping of the property-key parameters, use the "." notation instead (analogous to the notation of Java property files).

In **Application** objects, the configuration keys can be grouped by specifying them on the application level, type level (also referred to as class level), and instance level. Hence it is possible to attach configuration keys inside an **Application** object to FactoryTalk ProductionCentre buildtime instances, FactoryTalk ProductionCentre buildtime classes, or to the application itself.

TIP

Please keep in mind that the spelling of a key is case-sensitive and the key length depends on FactoryTalk ProductionCentre. For PharmaSuite, the length is <= 64 characters.

Configuration Keys of Phase Building Blocks

PharmaSuite provides the **Phase** type for configuration keys of phase building blocks. The phase itself corresponds to the instance level (e.g. ShowHistoricalDataChartPhase).

Example: Phase/ShowHistoricalDataChartPhase/HistoricalDataChartReportResolution

- Phase
  - ShowHistoricalDataChartPhase
    - HistoricalDataChartReportResolution

Lookup Mechanism

This section outlines the FactoryTalk ProductionCentre lookup mechanism for configuration keys in **Application** objects.

In the example below, the system performs the following steps when searching for the **Form/ConfKeyForm/key** configuration key.

1. Instance level (station)
   First, the system searches for the **Form/ConfKeyForm/key** key in the current station (here: **ConfKeyStation**).

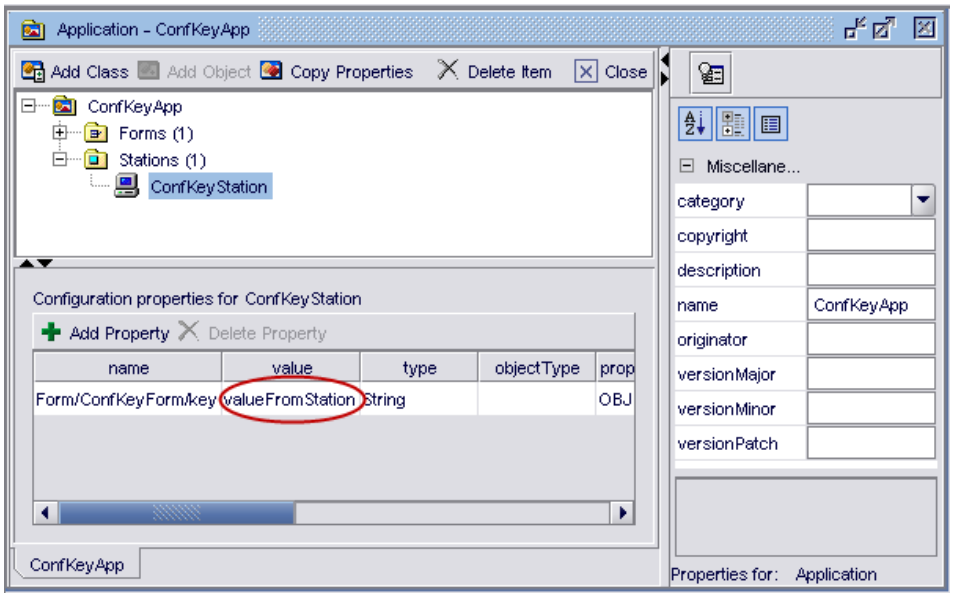

Figure 4: Lookup on instance level (station)

2. Instance level (form)
   If the search on the instance level (station) was not successful, the system searches in a particular instance of the **ConfKeyForm** form in the forms of the **Application** object (here: **ConfKeyApp**). If the key is specified in the form, we will receive its value.

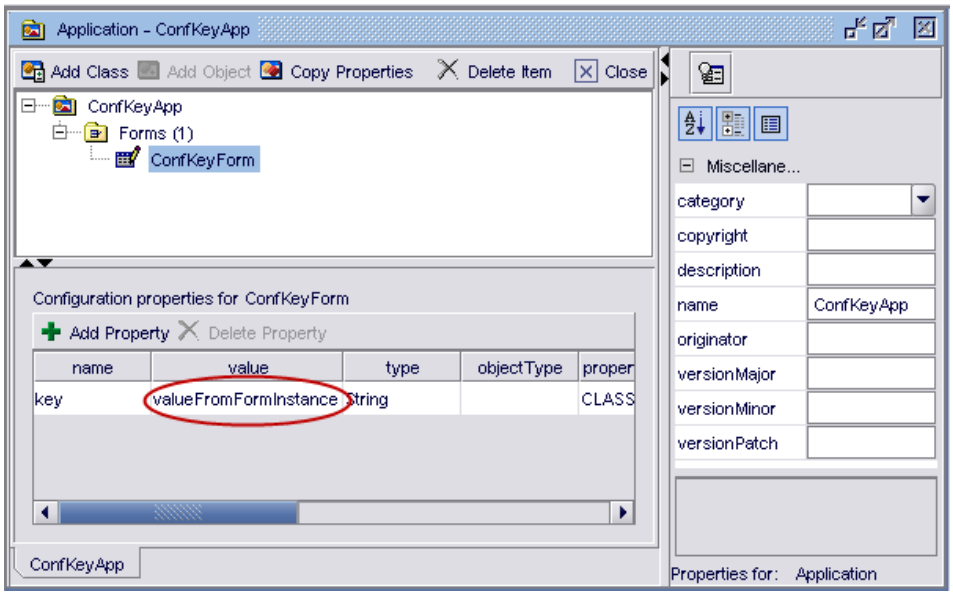

Figure 5: Lookup on instance level (form)

Class level
If we specified the key on class level, only the search on this level would yield a value due to the implicit default mechanism of FactoryTalk ProductionCentre.

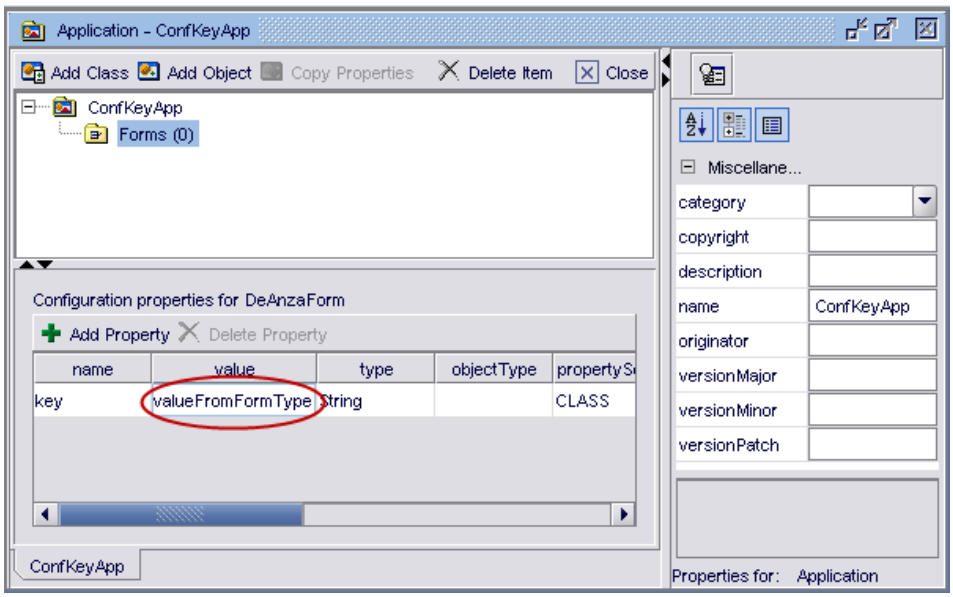

Figure 6: Lookup on class level

3. Application level
   If we still do not get a value for the **Form/ConfKeyForm/key** key, the system searches for the key in the configuration properties on the application level.

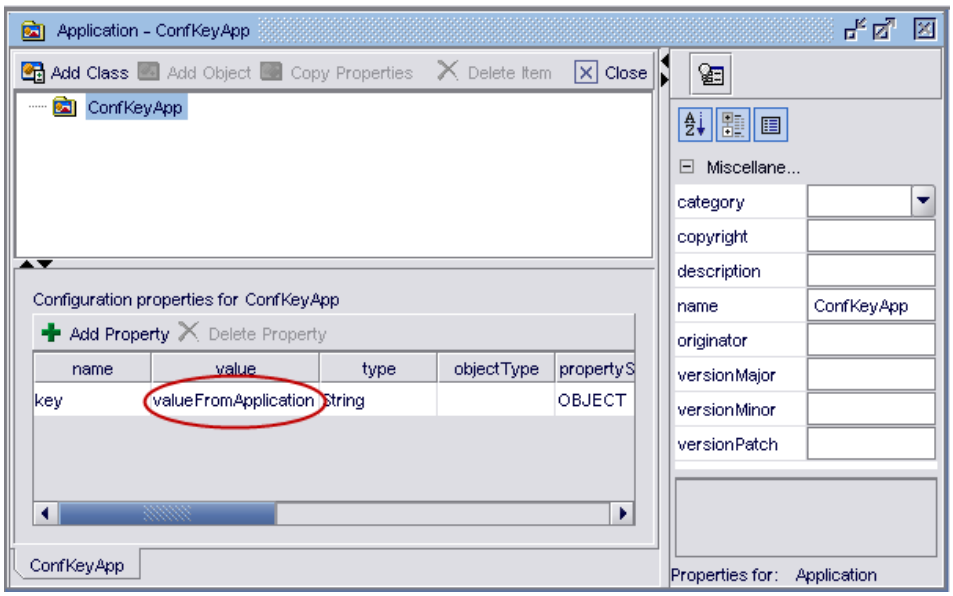

Figure 7: Lookup on application level

The steps (1.) to (3.) are repeated for all nested applications and their nested applications in the configuration hierarchy tree. If no value is found, the default value defined by the API is returned.

Supported Buildtime Objects

The following types of buildtime objects are supported as application configuration keys. Their names must be specified as the first part of the configuration key. Please consider the correct spelling given here (it might differ from the spelling of the names as displayed in Process Designer).

- AccessPrivilege
- Account
- ActivitySet
- ATDefinition
- BillOfMaterials
- Carrier
- CarrierClass
- Equipment
- EquipmentClass
- FlexibleStateModel

- Form
- Hierarchy
- LabelDataDefinition
- LabelDesign
- LibraryHolder
- List
- Locale
- Location
- MasterRecipe
- Messages
- Part
- PartClass
- ProcessBom
- ProductionLine
- ProductionQueue
- ReportDataDefinition
- ReportDesign
- ResourceCondition
- ResourceRoute
- Route
- SemanticProperty
- SemanticPropertySet
- Station
- TestDefinition
- User
- UserGroup
- WorkCenter
- WorkInstruction

Logging of Customer-specific Configuration Keys

Customer-specific configurations and their associated configuration keys that are defined on the customer level are logged in the default logging folder of PharmaSuite in the

Windows default user directory:
.../Documents and Settings/<User>/ApplicationData/Rockwell Automation/FactoryTalk ProductionCentre/Logs/PlantOpsClient/ApplicationConfigurationSettings.log.

This file includes not only the customer-specific configuration keys and their values, it also contains the application name, the application configuration keys, and their instances. The log file provides information at client start-up including the logged-in user, current station, work center, and the current date and time.

Dynamic Change of Configuration

Each time a user logs into a PharmaSuite application the configuration is loaded based on the current user and station configuration. However, it is possible to refresh the **MESConfiguration** configuration and notify listeners that are interested in the event. The API of the **MESConfiguration** configuration supports this feature with the following static methods:

- *addRefreshListener* allows to add a listener to be notified after the reload has taken place. It is possible to add multiple listeners.
- *removeRefreshListener* allows to remove a listener.
- *refresh* removes the existing configuration and reloads it from the middle tier based on the current user and station. After the reload, all registered listeners are notified.

A user or station change is probably the most common use case for a reload of the **MESConfiguration** configuration. However, keep in mind that most configuration keys are expected to be identical for all users and stations across the system. That leaves only very specific configuration keys, which may require your application or building block to register for configuration key changes. For details, see the description of the configuration keys in chapter "Configuration Keys of PharmaSuite" (page 29).

TIP

Do not store configuration items in fields of a class. In this case, you create a local cache, which can get out of sync with the current set of configuration items. Keep in mind that some other data like style sheet data is derived from the **MESConfiguration** configuration. If you need to use a cache, consider to register the class as a listener to the **MESConfiguration** refresh events.

Defining a Configuration Key

To define a configuration key, proceed as follows:

1. In Process Designer, expand the **Applications** node and open the relevant **Application** object.
2. In the details frame of the **Application** object, select the item whose configuration key you are defining.
   An item is the application, a class of the application, or an instance of a class.
3. If the configuration key does not exist:
   In the configuration properties frame, click the **Add Property** button and enter the required data.
4. If the configuration key already exits:
   In the configuration properties frame, select the cell whose value should be updated. Modify the value.
5. Save the **Application** object.

Overriding Configuration Keys in Unit Tests

When unit testing a method that depends on configuration settings, it is sometimes desirable to set the value of a configuration key without changing the database. A typical use case is: you are about to test a method that depends on a Boolean key and you probably wish to call the method twice from your unit tests, once for true and once for false. So your test will cover the tested code completely.

Use the *ConfigurationUnitTestHelper* class to perform a unit test in this way:

```
// preset a certain configuration key to false
ConfigurationUnitTestHelper.getCache().setBoolean(theConfigKey, false);

// call method to test which depends on "theConfigKey" and check its result
...
```

TIP

Tests that run in the same Java process will see the modifications made to the configuration cache. Thus, when you write a unit test it is recommended to check which configuration keys will be read during the test and to set the values for all these keys to a defined value upfront. This approach has the additional advantage that the unit test is not affected if the default value for such a configuration key changes in the database.

Chapter 4

Logging and Debugging

This section describes the configuration of the PharmaSuite logging capability and how it can be used for debugging purposes. We assume that the reader is familiar with the basic concept of log4j logging and the logging capabilities of FactoryTalk ProductionCentre in general.

Logging is typically used for the following reasons:

- to document certain events of the system. Typical events are startup, shutdown of an application, malfunction of specific function.
- to trace internal events for debugging purposes.

PharmaSuite is pre-configured to write logger events of the types mentioned under first bullet to a log file and the console. It does not automatically log other events. However, if you implement new functions or you wish to analyze a certain behavior in greater detail, you can use more elaborate log settings to have more information logged.

Configuring the Logging Capability

PharmaSuite logging capability uses **log4j**. For more information on the log4j package, please refer to Apache Logging Services Project [D1] (page 142).

PharmaSuite supports the configuration of log4j on different levels:

1. *log4j.properties*
   This is the default, fallback configuration property file. Unfortunately, this file is one of quite a number of files of a Java package. For this reason, we cannot depend on this file alone.
2. *log4j_ftps.properties*
   This is the PharmaSuite-specific property file. This file is part of the PharmaSuite delivery. It affects the logging behavior of all clients.
3. *log4j_custom.properties*
   This property file is not part of the PharmaSuite delivery. A system integrator can create this file to override the default PharmaSuite settings. This property file will be searched on the classpath. In order to make PharmaSuite find it, it is sufficient to create a JAR file containing the desired *log4j_custom.properties* file. Then load the JAR file as library into Process Designer. It affects the logging behavior of all clients.

4. *log4j_custom_local.properties*
   This property file is optional. If it is stored in the log directory of a running process, it affects only the behavior of this process.
   The Express Edition installation creates a specific property file for each PharmaSuite server. The PharmaSuite servers are running as Windows services.

   - Example: The EBR server has been started as a service on the server.
     log4j_custom_local.properties is stored in
     <installDirectory>\installation\services\PharmaSuite_EBR_Server\bin\logs
     where *<installDirectory>* is the default installation path on the server (*C:\Rockwell\PharmaSuite*).
     You can also create the file manually on the client.
     When the Enterprise Edition installation has been used, see section "Setting up Shop Operations Servers" in "Technical Manual Installation - Enterprise Edition" [A5] (page 141).
   - Example: A Shop Operations Client has been started in a browser on the client.
     log4j_custom_local.properties is stored in
     C:\.FTPC\<MES-PS-HOST>\ProductionCentre\logs\PlantOpsClient
     where *<MES-PS-HOST>* is the name of your PharmaSuite server.
   - Example: A Shop Operations Client has been started in a development environment under Windows 7.
     log4j_custom_local.properties is stored in *c:\Users\<Windows user>\AppData\Roaming\Rockwell Automation\FactoryTalk ProductionCentre\logs\PlantOpsClient*
     where *<Windows user>* is the logged-in Windows user.

TIP

The *log4j* property files of levels 1 through 3 are only loaded when they exist when PharmaSuite is started.
The loading behavior related to a *log4j_custom_local.properties* file is as follows: the creation and changes are dynamically watched every 10 seconds. If the file does not exist on service start, it is also possible to create the file later, when the service is running. This is useful to trace the behavior of services with different log switch settings and without the necessity to restart the service.

The output of the logger is written to the console and the PharmaSuite log file for a Shop Operations Client. The log file name consists of a client prefix and *ftps.log*. Example: *ApplicationStart_ProductionExecutionClient-ftps.log*.
The log file name of a Shop Operations Server consists of *<PharmaSuite event sheet name>-ftps.log*. Example: *PharmaSuite_EBR_Server-ftps.log*.
In the default configuration, the **log file for clients** (e.g. Shop Operations) is located under
C:\.FTPC\<MES-PS-HOST>\ProductionCentre\logs\PlantOpsClient\<prefix>-ftps.log.

This directory also contains all the other *PlantOpsClient*-specific log files. For more information, please refer to the FactoryTalk ProductionCentre configuration documentation.
Per default, the **log file for services** (e.g. EBR Server) is located under *<installDirectory>\installation\services\PharmaSuite_EBR_Server\bin\logs* where *<installDirectory>* is the default installation path on the server (*C:\Rockwell\PharmaSuite*).

All log4j properties files have the same structure. When there are several properties files defined, the most general settings, as stated in the *log4j.properties* file, are overridden by the more specific settings stated in the other properties files. Thus, settings of *log4j.properties* are overridden by settings of *log4j_ftps.properties*, which in turn can be overridden by settings defined in *log4j_custom.properties* and so forth. This means a setting in the *log4j_custom_local.properties* file overrides any setting in a more general log4j properties file.
An example of the PharmaSuite-specific *log4j_ftps.properties* (page 24) file is given below.

Configuring log4j Properties Files

When configuring the log4j properties files you can set various log levels and use loggers to further refine your configuration.

Log Levels

The following hierarchy applies to the log levels used by log4j:

ALL < TRACE < DEBUG < INFO < WARN < ERROR < FATAL < OFF

ALL is the most verbose setting. Each logger entry will be written to the log file. **OFF** is the least verbose setting, where no logger entry will be written to the log file. The **DEBUG** log level means that all logger entries with **DEBUG** or higher (**INFO**, **WARN**, **ERROR**, **FATAL**, **OFF**) will be written to the log file.

TIP

Due to a dependency to FactoryTalk ProductionCentre, currently the **TRACE** log level is only available to a limited extent. Therefore it will be mapped to the **DEBUG** log level. If **DEBUG** is your configured log level, also all **TRACE** messages are logged. Since the **TRACE** messages are logged with **DEBUG** as prefix, they cannot be distinguished from **DEBUG** logs.

Loggers

You can configure the effective log level for entire packages down to single Java source files. The pre-configured setting for all Java files being part of *com.rockwell* is **WARN**, which means that logger entries with **WARN** or higher will be written to the log file.

To switch on the **DEBUG** level for the *log4j.logger.com.rockwell.mes.clientfw.pec.ifc.helper.InOutputPropertiesOfForm* class, add the following line to the *log4j* properties file.

```
log4j.logger.com.rockwell.mes.clientfw.pec.ifc.helper.InOutputPropertiesOfForm=DEBUG
```

TIP

Please note the **log4j.logger** prefix that precedes the fully qualified class name.

If you wish this setting to be only temporary and to apply only to one specific station, add the line to the *log4j_custom_local.properties* file on the respective station.

SAMPLE CONFIGURATION FILE: LOG4J_FTPS.PROPERTIES

The default configuration of the PharmaSuite-specific *log4j_ftps.properties* file is as follows.

```
# This is the root configuring for logging
log4j.rootCategory=WARN, stdout, logfile

### direct log messages to stdout ###
log4j.appender.stdout=org.apache.log4j.ConsoleAppender
log4j.appender.stdout.Target=System.out
log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
log4j.appender.stdout.layout.ConversionPattern=%d{ISO8601} [%t] %5p %c{1}\:%L - %m%n

### log to file ###
log4j.appender.logfile=org.apache.log4j.RollingFileAppender
log4j.appender.logfile.File=${PCContext.logPath}/${PCContext.logFilePrefix}-ftps.log
log4j.appender.logfile.MaxFileSize=5MB
log4j.appender.logfile.MaxBackupIndex=10
log4j.appender.logfile.layout=org.apache.log4j.PatternLayout
log4j.appender.logfile.layout.ConversionPattern=%d{ISO8601} [%t] %5p %c{1}\:%L - %m%n

# If programmed properly the most messages would be at DEBUG and the least at FATAL.
log4j.logger.com.rockwell=WARN

# Control logging for other open source packages
log4j.logger.org.apache.commons=ERROR
```

The most important configuration properties are at the last two properties defined:

- This line indicates that all logger entries higher than or equal to **WARN** will be written to the log file.
  This setting affects all packages starting with *com.rockwell*.

```
log4j.logger.com.rockwell=WARN
```

- This line indicates that all logger entries higher than or equal to **ERROR** will be written to the log file.
  This setting affects all packages starting with *org.apache.commons*.

```
log4j.logger.org.apache.commons=ERROR
```

Debugging Support

This section describes some common use cases where it may be helpful to use special log settings for debug purposes.

Workflow Analysis

PharmaSuite uses activity sets to implement pre-defined workflows. If a specific workflow does not work as expected and even if you are developing a new workflow, you may wish to see which activity steps are executed and which data is being passed in and out of each activity step.

To feed this information to the log file, add the following line to your local *log4j_custom_local.properties* file:

```
log4j.logger.com.rockwell.mes.clientfw.pec.ifc.helper.InOutputPropertiesOfForm=DEBUG
```

If you start the **Material Receipt** workflow with this setting, the logging output will be similar to the sample log shown below. Each time a *FormActivity* (subclass of *BaseFormActivity*) is completed it dumps its output properties including the current values. The sample dump below shows a walk through the **Material Receipt** workflow

```
start thread=ActivitySet=GoodsReceipt/Step=asCheckStationIsSet
start thread=ActivitySet=GoodsReceipt/Step=asStepCreateTHContext
start thread=ActivitySet=as_wip_GoodsReceiptSelectPart/Step=asMarkUnfinished
start thread=activityControlThread
13:15:24,382 DEBUG InOutputPropertiesOfForm:159 - Form [pec_FilterPartsForm] OutputItems:
13:15:24,413 DEBUG InOutputPropertiesOfForm:164 - Output [filter] Value:[ WHERE  (  (  (
PART.part_number LIKE 'D001%' )  )  ) ]
start thread=activityControlThread
13:15:29,444 DEBUG InOutputPropertiesOfForm:159 - Form [pec_GoodsReceiptSelectPart]
OutputItems:
13:15:29,444 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_selectedPart]
Value:[D001-11.1]
13:15:29,444 DEBUG InOutputPropertiesOfForm:164 - Output [nextStep] Value:[1]
13:15:29,444 DEBUG InOutputPropertiesOfForm:164 - Output [orderStep] Value:[null]
13:15:29,444 DEBUG InOutputPropertiesOfForm:164 - Output [partFilter] Value:[ WHERE  (  (
( PART.part_number LIKE 'D001%' )  )  ) ]
13:15:29,444 DEBUG InOutputPropertiesOfForm:164 - Output [selectablePartsConstraint]
Value:[null]
start thread=ActivitySet=as_wip_GoodsReceiptSelectPart/Step=asMarkFinished
start thread=activityControlThread
13:15:35,226 DEBUG InOutputPropertiesOfForm:159 - Form [pec_GoodsReceiptCreateBatch]
OutputItems:
13:15:35,226 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_batch]
Value:[BX307]
13:15:35,226 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_selectedPart]
Value:[D001-11.1]
13:15:35,226 DEBUG InOutputPropertiesOfForm:164 - Output [orderStep] Value:[null]
start thread=ActivitySet=as_wip_GoodsReceiptSelectStorageLocation/Step=asMarkUnfinished
start thread=activityControlThread
13:15:43,616 DEBUG InOutputPropertiesOfForm:159 - Form [pec_FilterLocationsForm]
OutputItems:
13:15:43,647 DEBUG InOutputPropertiesOfForm:164 - Output [filter] Value:[ WHERE  (  (  (
LOCATION.location LIKE 'W%' )  AND  ( UDA_Location.X_hierarchy_I=0 )  )  ) ]
start thread=activityControlThread
13:15:45,382 DEBUG InOutputPropertiesOfForm:159 - Form
```

```
[pec_GoodsReceiptSelectStorageLocation] OutputItems:
13:15:45,382 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_batch]
Value:[BX307]
13:15:45,382 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_selectedLocation]
Value:[WD_Location]
13:15:45,382 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_selectedPart]
Value:[D001-11.1]
13:15:45,413 DEBUG InOutputPropertiesOfForm:164 - Output [locationFilter] Value:[ WHERE  (
(  ( LOCATION.location LIKE 'W%' )  AND  ( UDA_Location.X_hierarchy_I=0 )  )  ) ]
13:15:45,413 DEBUG InOutputPropertiesOfForm:164 - Output [selectableLocationsConstraint]
Value:[null]
start thread=ActivitySet=as_wip_GoodsReceiptSelectStorageLocation/Step=asMarkFinished
start thread=ActivitySet=GoodsReceipt/Step=asStepPrepareSublotCreation
start thread=activityControlThread
13:15:57,460 DEBUG InOutputPropertiesOfForm:159 - Form [pec_GoodsReceiptCreateSublots]
OutputItems:
13:15:57,460 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_batch]
Value:[BX307]
13:15:57,460 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_selectedLocation]
Value:[WD_Location]
13:15:57,460 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_selectedPart]
Value:[D001-11.1]
13:15:57,460 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_vectorSublots]
Value:[[SL00001284]]
13:15:57,460 DEBUG InOutputPropertiesOfForm:164 - Output [orderStep] Value:[null]
13:15:57,460 DEBUG InOutputPropertiesOfForm:164 - Output
[transactionHistoryClientContext] Value:[45c90ae2-a1ed-4c3e-a6dc-eb27c9df45b4]
start thread=activityControlThread
13:16:18,835 DEBUG InOutputPropertiesOfForm:159 - Form [pec_GoodsReceiptPrintLabels]
OutputItems:
13:16:18,835 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_batch]
Value:[BX307]
13:16:18,835 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_selectedLocation]
Value:[WD_Location]
13:16:18,835 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_selectedPart]
Value:[D001-11.1]
13:16:18,835 DEBUG InOutputPropertiesOfForm:164 - Output [GoodsReceipt_vectorSublots]
Value:[[SL00001284]]
13:16:18,835 DEBUG InOutputPropertiesOfForm:164 - Output [nextStep] Value:[null]
13:16:18,835 DEBUG InOutputPropertiesOfForm:164 - Output [orderStep] Value:[null]
```

Barcode Support

Barcode scanning is frequently used for various identification purposes. Software interprets a barcode scan operation as keyboard input. Each scan starts with a specific start key, followed by the barcode with its specific format, and ends with another specific end character.

To analyze how keyboard input is processed and how barcode scanning is being processed, switch on **DEBUG** logging for the *BarcodeListener*.

To feed this information to the log file, add the following line to your local *log4j_custom_local.properties* file:

```
log4j.logger.com.rockwell.mes.commons.base.ifc.BarcodeListener=DEBUG
```

If you start an EBR recipe like the following **Dispense**-related process with this setting, the logging output will be similar to the sample log shown below.

```
13:21:30,132 DEBUG BarcodeListener:643 - ignoreKeyEvents(false)
13:21:36,960 DEBUG BarcodeListener:643 - ignoreKeyEvents(true)
13:21:37,772 DEBUG BarcodeListener:643 - ignoreKeyEvents(false)
start thread=activityControlThread
13:21:37,819 DEBUG BarcodeListener:643 - ignoreKeyEvents(true)
13:21:46,866 DEBUG BarcodeListener:643 - ignoreKeyEvents(false)
13:21:53,147 DEBUG BarcodeListener:437 - Time: 1319541713147 ''
13:21:55,585 DEBUG BarcodeListener:437 - Time: 1319541715585 ''
13:22:01,819 DEBUG BarcodeListener:294 -
java.awt.event.KeyEvent[KEY_PRESSED,keyCode=524,keyText=Windows,keyChar=Undefined
keyChar,keyLocation=KEY_LOCATION_LEFT,rawCode=91,primaryLevelUnicode=0,scancode=91] on
ConfirmButton
13:22:02,335 DEBUG BarcodeListener:158 - startKey
13:22:02,335 DEBUG BarcodeListener:294 -
java.awt.event.KeyEvent[KEY_RELEASED,keyCode=115,keyText=F4,keyChar=Undefined
keyChar,keyLocation=KEY_LOCATION_STANDARD,rawCode=115,primaryLevelUnicode=0,scancode=62
] on ConfirmButton
13:22:02,335 DEBUG BarcodeListener:295 - scanActive=true isIgnoreKeyEvents=false
13:22:02,351 DEBUG BarcodeListener:294 -
java.awt.event.KeyEvent[KEY_PRESSED,keyCode=16,keyText=Shift,keyChar=Undefined
keyChar,modifiers=Shift,extModifiers=Shift,keyLocation=KEY_LOCATION_LEFT,rawCode=16,pri
maryLevelUnicode=0,scancode=42] on ConfirmButton
```

Focus Issues Analysis

Production Execution Client processes and workflows are designed to be operated without a pointing device such as a mouse. This requires the application to be sensitive to focus settings. To analyze this, you can switch on logging of the tabulation sequence when entering a **FormActivity**. In Process Designer, enable the developer-mode for the user you use and, in Process Designer, start the Production Execution Client. Press the CTRL+F2 key combination to get the current focus owner logged.

To feed this information to the log file, add the following line to your local *log4j_custom_local.properties* file:

```
log4j.logger.com.rockwell.mes.clientfw.commons.ifc.helper.MesTabulationSequence=DEBUG
```

If you start the **Material Receipt** pre-defined workflow with this setting, the logging output will be similar to the sample log shown below. The tabulation sequence is printed after entering the **Filter Materials** step. In addition, the CTRL+F2 key combination was pressed once to display the current focus owner.

```
start thread=ActivitySet=GoodsReceipt/Step=asCheckStationIsSet
start thread=ActivitySet=GoodsReceipt/Step=asStepCreateTHContext
start thread=ActivitySet=as_wip_GoodsReceiptSelectPart/Step=asMarkUnfinished
start thread=activityControlThread
13:46:50,554 DEBUG MesTabulationSequence:279 - 0. pec_FilterPartsForm(Form)
13:46:50,554 DEBUG MesTabulationSequence:261 -   1. 0 : panelData(Panel)
13:46:50,554 DEBUG MesTabulationSequence:261 -     1. 27 : groupboxBasicData(GroupBox)
13:46:50,554 DEBUG MesTabulationSequence:261 -       1. 10 : comboboxPartName(ComboBox)
13:46:50,554 DEBUG MesTabulationSequence:261 -       2. 15 : editPartName(Edit)
13:46:50,554 DEBUG MesTabulationSequence:261 -       3. 20 :
comboboxPartDescription(ComboBox)
13:46:50,554 DEBUG MesTabulationSequence:261 -       4. 25 : editPartDescription(Edit)
13:46:50,585 DEBUG MesTabulationSequence:261 -       5. 30 : comboboxPartType(ComboBox)
13:46:50,601 DEBUG MesTabulationSequence:261 -       6. 36 : labelMaxRows(FlatLabel)
13:46:50,601 DEBUG MesTabulationSequence:261 -       7. 37 : smartEditMaxRows(SmartEdit)
```

```
13:46:50,601 DEBUG MesTabulationSequence:261 -       8. 38 :
activityControlPartType(ChoicelistFieldActivity)
13:46:50,601 DEBUG MesTabulationSequence:261 -         1. 0 : comboBox(ComboBox)
13:46:50,601 DEBUG MesTabulationSequence:261 -     2. 28 : groupboxSort(GroupBox)
13:46:50,601 DEBUG MesTabulationSequence:261 -       1. 100 : comboboxSort(ComboBox)
13:49:58,147 DEBUG MesTabulationSequence:223 - Focus:
com.datasweep.compatibility.ui.Edit$EditTextField[,0,0,580x25,layout=javax.swing.plaf.b
asic.BasicTextUI$UpdateHandler,alignmentX=0.0,alignmentY=0.0,border=javax.swing.border.
CompoundBorder@159a197,flags=296,maximumSize=,minimumSize=,preferredSize=java.awt.Dimen
sion[width=75,height=25],caretColor=javax.swing.plaf.ColorUIResource[r=0,g=0,b=64],disa
bledTextColor=java.awt.Color[r=100,g=100,b=100],editable=true,margin=javax.swing.plaf.I
nsetsUIResource[top=2,left=3,bottom=2,right=2],selectedTextColor=javax.swing.plaf.Color
UIResource[r=0,g=0,b=64],selectionColor=javax.swing.plaf.ColorUIResource[r=158,g=191,b=
227],columns=0,columnWidth=0,command=,horizontalAlignment=LEFT](Edit$EditTextField)
```

Configuring Logging for PharmaSuite Servers

PharmaSuite servers run as a Windows service and use messaging to communicate with a PharmaSuite client (e.g. Production Execution Client, Production Response Client).
To change or configure the log levels related to a server, use the *<installDirectory>\installation\services\PharmaSuite_<serverName>\bin\logs\log4j_custom_local.properties* file, where *<installDirectory>* is the default installation path on the server (*C:\Rockwell\PharmaSuite*) and *<serverName>* is the name of the PharmaSuite server (e.g. *EBR_Server*).

For each event sheet (of a PharmaSuite server), PharmaSuite provides a specific *\log4j_custom_local.properties* file with meaningful log switches.

```
-- Customer specific log property file for EBR Server
--
-- REMARK: This file is watched on changes. After about 10 seconds they take effect.
--
-- EBR Server
log4j.logger.com.rockwell.mes.apps.ebr.ifc.activities=INFO
--
-- Messaging, logged in a separate file by PharmaSuite default
log4j.logger.com.rockwell.mes.commons.messaging.ifc=INFO
```

TIP

Be aware, that changes made to this file take effect about 10 seconds after the change without restarting the service.

If a switch was dynamically set to a given value during a running process, it keeps its last state until you change it.
To change the value of an active switch (e.g. INFO), explicitly set it to a given value (ERROR) and do not comment the line, because this will have no effect on a running process.

Chapter 5

Configuration Keys of PharmaSuite

This section describes the configuration keys of PharmaSuite. They are listed by application area. For general information on configuration keys, see chapter "Managing Configurations" (page 11).

A configuration key is evaluated

- only when a PharmaSuite client is started or
- also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.

TIP

Configuration keys are modified and defined in Process Designer.

Behavior of empty values:

- PharmaSuite treats a configuration key with an empty value like a non-existing configuration key. That means the default value of the configuration key is used, not the null value.
- To overwrite the default value of a configuration key of the **String** type, set its value to **<RA_EMPTY_STRING>**.

Global Configuration Keys

The global configuration keys describe and configure the basic behavior of the system.

Form/ComponentOrientation/RightToLeft

- **Type**: Boolean
- **Value**: False
- **Description**: Defines the orientation of PharmaSuite related to left-to-right and right-to-left language features, respectively.
  If the value is set to **true**, the right-to-left orientation is enabled and applied to the layout of PharmaSuite.
  If the value is set to **false**, the left-to-right orientation is enabled and applied to the layout of PharmaSuite. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/commons-base-ifc.jar/BarcodeListener.endKey

- **Type**: Long
- **Value**: java.awt.event.KeyEvent.VK_ENTER (carriage return)
- **Description**: Defines the suffix key code for the barcode scanning. It is in a form of a constant from the Java KeyEvent class, rather than an ASCII code. If this constant is changed, all barcode scanning devices have to be reconfigured.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: See **BarcodeListener.startKey** configuration key (page 30).

LibraryHolder/commons-base-ifc.jar/BarcodeListener.startKey

- **Type**: Long
- **Value**: java.awt.event.KeyEvent.VK_F4 (function key F4)
- **Description**: Defines the prefix key code for the barcode scanning. It is in a form of a constant from the Java KeyEvent class, rather than an ASCII code. If this constant is changed, all barcode scanning devices have to be reconfigured.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A key, which can be sent by a desired scanner (see the manual of the scanner). The key must not interfere with any of the accelerator keys used for the user interface of the Production Execution Client.

LibraryHolder/commons-base-ifc.jar/BarcodeListener.timeout

- **Type**: Long
- **Value**: 1000 (i.e. one second)
- **Description**: Defines the timeout for the barcode scanning in milliseconds (zero means no timeout). This is the maximum time span between receiving the start key and receiving the end key of a single. If more time elapses, the scanning is considered invalid and aborted. The value of **0** means that no timeout feature is active and the barcode scan may last incessantly.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [0... maxInt]

LibraryHolder/commons-base-ifc.jar/calculateScaleForPercentage

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, the scale of the result will be newly calculated on multiplications with the percentage measured values. The new scale

will be calculated using the following formula:
ROUND_CEILING(scale_of_measured_value - log10(value_of_percent_multiplicator) + log10(100))
If the value is set to **false**, the scale of the measured value will not be changed.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

LibraryHolder/commons-base-ifc.jar/calScaleDivMinimumPrecisionOffset

- **Type**: Long
- **Value**: 0
- **Description**: Defines the offset scale for division results of measured values (MV). The scale is defined by the system, and then extended by the amount of this configuration.
  Example:
  - **Definition**: **MVcalc** = 1.234; **MVscale** = 2; **MinimumPrecisionOffset** = 3
  - **Result**: **MVresult** = 1.23400
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/commons-base-ifc.jar/DeviceIdentificationPatternTerminalServices

- **Type**: String
- **Value**: %CLIENTNAME%
- **Description**: Specifies the pattern used to generate a unique device ID with environment variables when Terminal Services is used.
  The device ID is used to identify the physical device running a PharmaSuite application.
  By default the device ID is assigned to the value of the COMPUTERNAME environment variable.
  When Terminal Services is used (the SESSIONNAME environment variable exists and is not equal to Console), COMPUTERNAME is not unique. Then PharmaSuite uses this configuration key to calculate the device ID. If the default value is not unique, other information like USERNAME or SESSIONNAME can be used.
  Environment variables must begin with % and end with %.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: A concatenation of strings and existing environment variable names defined between %.

LibraryHolder/commons-base-ifc.jar/ServicesConfigFile

- **Type**: String
- **Value**: config.xml
- **Description**: Defines the services configuration file. The default value points to the *config.xml* file, which is included in the PharmaSuite standard. It includes references to all XML files that define service configurations. If you wish to add new services or modify existing services this configuration key has to be changed to the new adapted services configuration file. We recommend to adapt the original *config.xml* file and rename it. The adapted file itself must be located in the Java classpath and must not collide with existing file names.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

MandatoryServerEventSheets

- **Type**: String - List
- **Value**: PharmaSuite_AI_Server:AutomationIntegrationServer1
  PharmaSuite_EBR_Server
  PharmaSuite_OE_Server
  PharmaSuite_TOM_Server
  PharmaSuite_Transition_Server
- **Description**: List of mandatory PharmaSuite servers that are relevant to:
  - Heartbeat monitoring of the Production Execution Client.
  - Production Execution Client startup checks: OE server and TOM server checks are only performed if configured. The EBR server is always checked.
  - Production Execution Client startup: ETOs are loaded only if TOM server is configured.
  - About dialogs: Status information of OE, TOM, and AI servers is only displayed if configured. Status information of the EBR server is always displayed.

  Remove servers from the list that are not used or needed, e.g. if only Dispense is used. The EBR server is always mandatory, so removing it has no effect.

Add servers to the list if additional AI server are available and must be checked or if additional Shop Operations Servers must be checked, e.g. if EIG is applied.

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Available PharmaSuite server event sheets that support server heartbeat.

TrustedPackages

- **Type**: String
- **Value**: java,com.rockwell,org.apache.activemq,com.datasweep,sun,com.osisoft
- **Description**: This configuration key is coupled with the security mechanisms used by ActiveMQ. For details, see ActiveMQ - Features - Message Features - ObjectMessage [D3] (page 142).
  List of packages that are whitelisted in order to be used in serialized object messages. This applies to the classes of the packages and their sub-packages.
  In order to bypass the security mechanism, i.e. to allow all classes, set the value to the * wildcard character.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

UoMsOfPotency

- **Type**: Object - List
- **Value**: UoMsOfPotency
- **Description**: List of units of measure that can be used for potency.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Configuration Keys for User Interface

The following configuration keys are available for configuring the user interface of both the Production Management Client and Production Execution Client. They are distinguished by their respective prefixes.

Form/acceleratorKeysMessagePack

- **Type**: String
- **Value**: ui_AcceleratorKeys
- **Description**: The message pack used to store the general accelerator keys.
- **Evaluated**: Only the accelerator keys not stored in a Production Management- or Execution-specific message pack will be taken from the general message pack.

- **Range**: N/A

Form/ApplicationStart/StyleSheet

- **Type**: Object - List
- **Value**: aps_StyleSheet
- **Description**: Defines the style sheet used in the PharmaSuite welcome page (application start).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ExceptionDashboard/StyleSheet

- **Type**: Object - List
- **Value**: prc_StyleSheet
- **Description**: Defines the style sheet used in the Production Response Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ProductionExecutionClient/acceleratorKeysMessagePack

- **Type**: String
- **Value**: pec_AcceleratorKeys
- **Description**: The message pack used to store the accelerator keys for the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ProductionExecutionClient/AlternateStyleSheet

- **Type**: Object - List
- **Value**: pec_StyleSheet_Alternate
- **Description**: Defines the style sheet used in the Production Execution Client for activity sets of the **ActivitySetNamesUsingAlternateStyleSheet** list.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Form/ApplicationStart_ProductionExecutionClient/ApplicationStart_MainIcon

- **Type**: String
- **Value**: pec_MainIcon
- **Description**: Contains the (base) name of the image to be displayed as the program icon for the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ProductionExecutionClient/doExitOnLogout

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, the Production Execution Client will be closed after the logout. If the value is set to **false**, the logout will be performed, but the application will stay open.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

Form/ApplicationStart_ProductionExecutionClient/hideMenu

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, the Production Execution Client will hide the FactoryTalk ProductionCentre file menu.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/ApplicationStart_ProductionExecutionClient/onlineHelpPrefix

- **Type**: String
- **Value**: /PharmaSuite/documentationandhelp/
- **Description**: The prefix is used to locate help pages for the Production Execution Client.
  The URL consists of three parts: server name, this prefix, and page name taken from a message pack.

Example:
http://ServerName.com:port/ThisPrefix/NameOfThePage.html, or
http://localhost:8080/PharmaSuite/documentationandhelp/index.htm.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Form/ApplicationStart_ProductionExecutionClient/StyleSheet

- **Type**: Object - List
- **Value**: pec_StyleSheet
- **Description**: Defines the style sheet used in the Production Execution Client.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Form/ApplicationStart_ProductionManagementClient/acceleratorKeysMessagePack

- **Type**: String
- **Value**: pmc_AcceleratorKeys
- **Description**: The message pack used to store the accelerator keys for the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ProductionManagementClient/ApplicationStart_MainIcon

- **Type**: String
- **Value**: pmc_MainIcon
- **Description**: Contains the (base) name of the image to be displayed as the program icon for the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ProductionManagementClient/doExitOnLogout

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, the Production Management Client will be closed after the logout. If the value is set to **false**, the logout will be performed, but the application will stay open.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/ApplicationStart_ProductionManagementClient/hideMenu

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, the Production Management Client will hide the FactoryTalk ProductionCentre file menu.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/ApplicationStart_ProductionManagementClient/onlineHelpPrefix

- **Type**: String
- **Value**: /PharmaSuite/documentationandhelp/
- **Description**: The prefix is used to locate help pages for the Production Management Client.
  The URL consists of three parts: server name, this prefix, and page name taken from a message pack.
  Example:
  http://ServerName.com:port/ThisPrefix/NameOfThePage.html, or
  http://localhost:8080/PharmaSuite/documentationandhelp/index.htm.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ProductionManagementClient/StyleSheet

- **Type**: Object - List
- **Value**: pmc_StyleSheet
- **Description**: Defines the style sheet used in the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.

- **Range**: N/A

Form/doExitOnLogout

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, the system will be closed after the logout. If the value is set to **false**, the logout will be performed, but the application will stay open.
- **Evaluated**: Only if the Production Execution- or Management-specific flag is not set.
- **Range**: [False, True]

Form/hideMenu

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, the system will hide the FactoryTalk ProductionCentre file menu.
- **Evaluated**: Only if the Production Execution- or Management-specific flag is not set.
- **Range**: [False, True]

Form/mes_DialogMessageBox

- **Type**: String
- **Value**: pmc-DialogMessageBox
- **Description**: Configures the form name of the standard dialog box, default is **pmc-DialogMessageBox**.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Form/StyleSheet

- **Type**: Object - List
- **Value**: N/A
- **Description**: Defines the style sheet used for the general application start form.
- **Evaluated**: Only when a PharmaSuite client is started.

- **Range**: N/A

Forms/<formName>/configureTabSequence

- **Type**: Boolean
- **Value**: False for forms of the Production Execution Client, True for forms of the Production Management Client
- **Description**: Defines whether the TAB sequence of the specified form (=<formName>) is evaluated automatically (from left to right and top to bottom) or whether the form designer has to maintain the TAB sequence manually.
  The TAB sequence is the order in which the focus moves through the UI controls of a form when you press the TAB key (without using the mouse).
  If the value is set to **true**, the TAB sequence is evaluated automatically.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

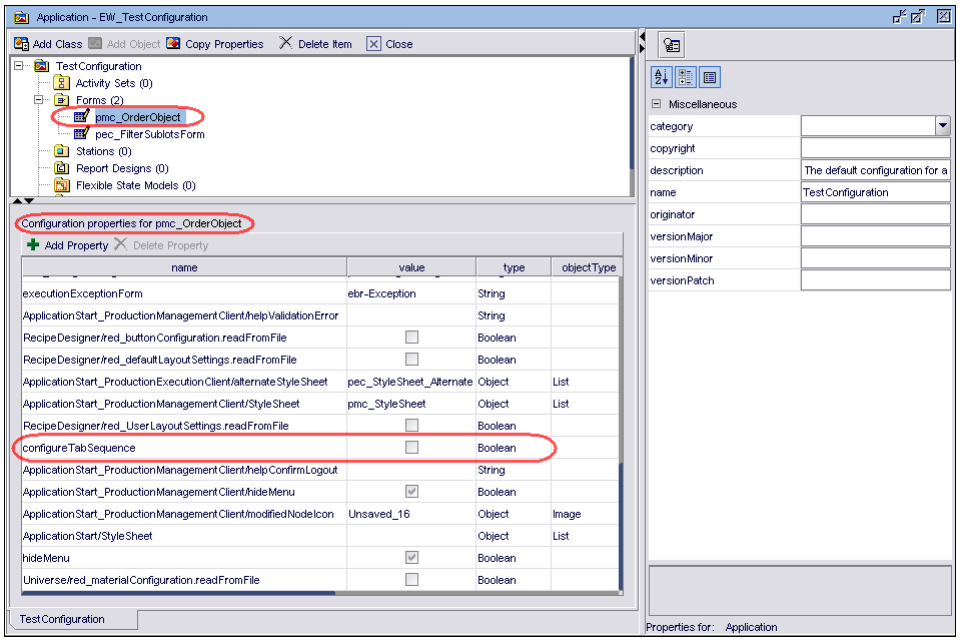

Figure 8: Disable automatically evaluated TAB sequence of the pmc_OrderObject form

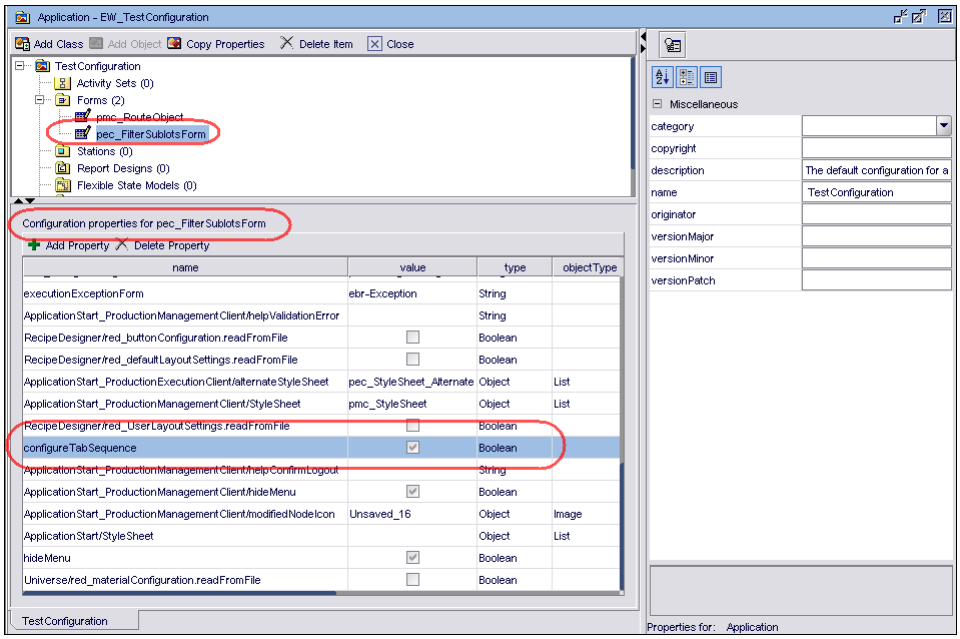

Figure 9: Enable automatically evaluated TAB sequence of the pec_FilterSublotsForm form

Station/hasTouchScreen

- **Type**: Boolean
- **Value**: True
- **Description**: Defines whether the station is equipped with a touch screen. If the value is set to **true**, the GridActivity doubles the row height.
- **Evaluated**: When a grid is refreshed in the Production Execution Client (e.g. a phase is switches from the **Preview** to the **Active** status).
- **Range**: [False, True]

Configuration Keys Specific to the Production Management Client

The following configuration keys are available for the Production Management Client.

Form/ApplicationStart_ProductionManagementClient/lockedObjectIcon

- **Type**: Object - Image
- **Value**: Locked_16
- **Description**: Defines the icon that is displayed in the tree view for all use cases to indicate the objects, which are being currently locked by other users for edition.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ProductionManagementClient/modifiedNodeIcon

- **Type**: Object - Image
- **Value**: Unsaved_16
- **Description**: Defines the icon that is displayed in the tree view for all use cases to indicate the objects, which have been modified and have not been saved yet.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_ProductionManagementClient/StartupChecks

- **Type**: List
- **Value**: N/A
- **Description**: List of configurable checks to be executed when the Production Management Client is started.
  Each check must implement the *IStartupCheck* interface.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for a **List** object in Process Designer.

Form/objectLockingRefreshInterval

- **Type**: Long
- **Value**: 10000
- **Description**: Defines the interval in milliseconds, after which the system rechecks if the objects that have been locked by other users are no longer locked.
- **Additional information**: This setting applies only to the displayed locks that have been released. It does not apply to new ones that have appeared. The parameter is checked, when a user selects an object.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: 1000 - ∞
  Be aware that applying a lower number will result in creating more database queries sent by each client automatically in the background. If you have many clients you should avoid lowering the default value.

LibraryHolder/services-discrete-impl.jar/DefaultDeviceNumberPrefix

- **Type**: String
- **Value**: SN
- **Description**: Defines the default prefix for serial numbers of devices.

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Two characters.

LibraryHolder/services-discrete-impl.jar/DefaultDeviceStorageLocation

- **Type**: Object - Location
- **Value**: MESDevice-SL
- **Description**: Defines the default storage location of devices.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any storage location.

LibraryHolder/services-discrete-impl.jar/DeviceNumberLeadingZeros

- **Type**: Boolean
- **Value**: False
- **Description**: Specifies if the numeric part of serial numbers of devices has leading zeros.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-discrete-impl.jar/DeviceNumberLength

- **Type**: Long
- **Value**: 20
- **Description**: Defines the maximum length of serial numbers of devices (including the prefix).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [9..32]

LibraryHolder/services-s88-impl.jar/SFCForceTransitionTimeout

- **Type**: Long
- **Value**: 3
- **Description**: Defines the timeout in seconds the system waits for a reply to a force execution transition request.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: > 0

LibraryHolder/services-s88-impl.jar/SFCTestForceTransitionTimeout

- **Type**: Long
- **Value**: 3
- **Description**: Defines the timeout in seconds the system waits for a reply to a test force execution transition request.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: > 0

Configuration Keys Specific to the Production Execution Client

The following configuration keys are available for the Production Execution Client to configure the behavior of workflows, also referred to as activity sets.

ActivitySet/[ActivitySetName]/allowMultipleInstances

- **Type**: Boolean
- **Value**: N/A
- **Description**: Allows the activity set with an **[ActivitySetName]** name to be started multiple times from the same client. If this property exists, it overwrites the **ActivitySet/allowMultipleInstances** general property (page 45) for this activity set. This also allows you to disable the feature for selected activity sets.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

ActivitySet/[ActivitySetName]/allowResumeOnlySameStation

- **Type**: Boolean
- **Value**: N/A
- **Description**: Allows the activity set with the **[ActivitySetName]** name to be resumed only at the station, at which it has been started. If this property exists, it overwrites the **ActivitySet/allowResumeOnlySameStation** general property (page 45) for this activity set.
  If the value is set to **false**, the Production Execution Client must be restarted in order to enable resuming at another station.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

ActivitySet/[ActivitySetName]/allowResumeOnlySameUser

- **Type**: Boolean
- **Value**: N/A
- **Description**: Allows the activity set with the **[ActivitySetName]** name to be resumed only by the user who has started it. If this property exists, it overwrites the **ActivitySet/allowResumeOnlySameUser** general property (page 45) for this activity set.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

ActivitySet/ActivitySetAutostartList

- **Type**: Object - List
- **Value**: AutostartActivitySetNames
- **Description**: Defines which activity sets will be started automatically when a user logs in.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

ActivitySet/AdministrativeTasks

- **Type**: String - List
- **Value**: (ReprintSimpleDispensingLabelAction, ReprintDispensingLabelAction, ReprintInventoryLabelAction, LogoutAction)
- **Description**: List of tasks for the **Administrative** task group.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Each string is a bean name that refers to a Java action class. Each action class must be extended by javax.swing.Action. The actions are instantiated through the PharmaSuite service factory.

ActivitySet/allowMultipleInstances

- **Type**: Boolean
- **Value**: False
- **Description**: If the value is set to **true**, it allows activity sets to be started multiple times from the same client.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

ActivitySet/allowResumeOnlySameStation

- **Type**: Boolean
- **Value**: True
- **Description**: Allows activity sets to be resumed only at the station, at which they have been started.
  If the value is set to **false**, the Production Execution Client must be restarted in order to enable resuming at another station.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

ActivitySet/allowResumeOnlySameUser

- **Type**: Boolean
- **Value**: False
- **Description**: Allows activity sets to be resumed only by the user who has started them.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

ActivitySet/StartableItemGroupEbrExecution

- **Type**: String
- **Value**: StartableItemNamesProduction

- **Description**: Defines the name of the list of the group in which operations of the EBR execution shall be displayed.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

ActivitySet/StartableItemGroups

- **Type**: Object - List
- **Value**: StartableItemGroups
- **Description**: Defines the startable activity sets/actions, their group and order in the Cockpit. The Cockpit contains startable workflows and actions, which are grouped by task groups. For each group exists a list, which contains the workflows and actions of that group. The elements of the list will be displayed in exact the same order, in which they are configured. The value of this parameter references a list containing the names of the lists for each group. The groups will be displayed in exactly the same order in which they are listed.
  If the name of a group contains the string **Invisible**, the group will not be displayed in the Cockpit, but nevertheless the actions defined for the group are active. This is useful for scanner actions.
  Actions are all names, which ends with the string **Action** e.g. **LogoutAction**. All other names are expected to be names of activity sets. This parameter defines which activity sets and actions will initially appear in the Cockpit, but further filtering of entries to be displayed can be done by **AccessPrivileges** and **UserGroups** settings.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

ActivitySet/StartableItemStartCampaignWeighing

- **Type**: String
- **Value**: StartCampaignWeighingAction
- **Description**: The startable item to start campaign weighing in the Production Execution Client.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: The startable item must be contained in any list of startable items.

ActivitySet/StartableItemStartDeviceProcessing

- **Type**: String
- **Value**: DeviceProcessingAction
- **Description**: The startable item to start device processing in the Production Execution Client.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: The startable item must be contained in any list of startable items.

ActivitySet/StartableItemStartRecipeWeighing

- **Type**: String
- **Value**: StartProcessingAction
- **Description**: The startable item to start batch processing in the Production Execution Client.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: The startable item must be contained in any list of startable items.

Form/ApplicationStart_ProductionExecutionClient/ChangeUserOrStationChecks

- **Type**: List
- **Value**: pec_ChangeUserOrStationChecks
- **Description**: List of configurable checks to be executed when the Change user or Register at station actions of the Production Execution Client are executed. Each check must implement the *IChangeUserOrStationCheck* interface.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for a **List** object in Process Designer.

Form/ApplicationStart_ProductionExecutionClient/NameOfCustomerLogo

- **Type**: String
- **Value**: ra_CustomerLogo
- **Description**: Image name of customer logo that is to be displayed on the left side of the status bar (header bar). Maximum size is 140 x 77 pixels.
- **Evaluated**: Only when a PharmaSuite client is started.

- **Range**: Any valid image name of an image present in the system.

Form/ApplicationStart_ProductionExecutionClient/StartupChecks

- **Type**: List
- **Value**: pec_StartupChecks
- **Description**: List of configurable checks to be executed when the Production Execution Client is started.
  Each check must implement the *IStartupCheck* interface.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for a **List** object in Process Designer.

Form/executionBatchRecordForm

- **Type**: String
- **Value**: ebr-BatchRecord
- **Description**: The name of the sub-form serving as the batch record form in the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/executionCockpitExceptionForm

- **Type**: String
- **Value**: ebr-CockpitException
- **Description**: The name of the sub-form serving as the Exception form within the Cockpit of the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/executionCockpitForm

- **Type**: String
- **Value**: ebr-Cockpit
- **Description**: The name of the sub-form serving as the Cockpit form in the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/executionExceptionForm

- **Type**: String
- **Value**: ebr-Exception
- **Description**: The name of the sub-form serving as the Exception form in the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/executionNavigationForm

- **Type**: String
- **Value**: ebr-Navigator
- **Description**: The name of the sub-form serving as the Navigator form in the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/executionPhaseActionForm

- **Type**: String
- **Value**: ebr-PhaseAction
- **Description**: The name of the sub-form serving as execution window form in the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/executionStartCampaignWeighingForm

- **Type**: String
- **Value**: ebr-StartCampaignWeighing
- **Description**: The name of the sub-form serving as the form to start dispensing the materials of several order steps together as campaign weighing in the Production Execution Client.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/clientfw-commons.ifc.jar/StandardMouseTouchScrollingBehavior

- **Type**: Boolean
- **Value**: True
- **Description**: In the default configuration (**true**), the content of a scroll pane within a phase UI is moved in the same direction as the cursor when the scrollbar is moved with a mouse button or a touch gesture. This scrolling mode is indicated by a special cursor.
  To enable moving scrollbars inside a phase panel, set the value **false**. In this configuration, the content of the area within the **inner** scrollbar is moved in the opposite direction when the mouse button moves the **inner** scrollbar.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/clientfw-pec-ifc.jar/AutomaticLockTimeoutInSeconds

- **Type**: Long
- **Value**: 900 (=15 min)
- **Description**: Defines the timeout in seconds before the Production Execution Client is automatically locked due to user inactivity. If the value is 0 or negative, the automatic lock feature is disabled.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any number.

LibraryHolder/services-wd-impl.jar/<usageType>.AutoConsumeStandaloneTargetSublot

- **Type**: Boolean
- **Value**: N/A
- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  If the value is set to **true**, the target sublots of a stand-alone weighing operation are consumed after the order step is completed.
  If the value is set to **false**, they are not consumed.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

LibraryHolder/services-wd-impl.jar/<usageType>.DispensingLabel

- **Type**: Object - ReportDesign
- **Value**: MESOrderRelatedSublotLabelRD
- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  The report design of the dispensing label.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre.

LibraryHolder/services-wd-impl.jar/<usageType>.LabelCount4TargetSublot

- **Type**: Long
- **Value**: 1
- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  Number of dispensing or pallet labels to be printed by the **Weigh** phase for the target sublots.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: >= 0

LibraryHolder/services-wd-impl.jar/<usageType>.Main

- **Type**: Object - ReportDesign
- **Value**: PS-BatchReport-Main
- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  The report design of the batch report.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

LibraryHolder/services-wd-impl.jar/<usageType>.PalletContainerSublotLabel

- **Type**: Object - ReportDesign
- **Value**: MESOrderRelatedSublotLabelPalletRD
- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  The report design of the pallet or container sublot label.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre.

LibraryHolder/services-wd-impl.jar/CostCenter.AutoConsumeStandaloneTargetSublot

- **Type**: Boolean
- **Value**: False
- **Description**: Applies to the **Cost center** usage type:
  If the value is set to **true**, the target sublots of a stand-alone weighing operation

are consumed after the order step is completed.
If the value is set to **false**, they are not consumed.
See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

LibraryHolder/services-wd-impl.jar/CostCenter.DispensingLabel

- **Type**: Object - ReportDesign
- **Value**: MESCostCenterRelatedSublotLabelRD
- **Description**: Applies to the **Cost center** usage type:
  The report design of the dispensing label.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre.

LibraryHolder/services-wd-impl.jar/CostCenter.LabelCount4TargetSublot

- **Type**: Long
- **Value**: 1
- **Description**: Applies to the **Cost center** usage type:
  Number of dispensing or pallet labels to be printed by the **Weigh** phase for the target sublots.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: >= 0

LibraryHolder/services-wd-impl.jar/CostCenter.Main

- **Type**: Object - ReportDesign
- **Value**: PS-BatchReport-Main
- **Description**: Applies to the **Cost center** usage type:
  The report design of the batch report.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

LibraryHolder/services-wd-impl.jar/CostCenter.PalletContainerSublotLabel

- **Type**: Object - ReportDesign
- **Value**: MESCostCenterRelatedSublotLabelPalletRD
- **Description**: Applies to the **Cost center** usage type:
  The report design of the pallet or container sublot label.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre.

LibraryHolder/services-wd-impl.jar/Production.DispensingLabel

- **Type**: Object - ReportDesign
- **Value**: MESOrderRelatedSublotLabelRD

- **Description**: Applies to the **Production** usage type:
  The report design of the dispensing label.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre.

LibraryHolder/services-wd-impl.jar/Production.Main

- **Type**: Object - ReportDesign
- **Value**: PS-BatchReport-Main
- **Description**: Applies to the **Production** usage type:
  The report design of the batch report.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

LibraryHolder/services-wd-impl.jar/Production.PalletContainerSublotLabel

- **Type**: Object - ReportDesign
- **Value**: MESOrderRelatedSublotLabelPalletRD
- **Description**: Applies to the **Production** usage type:
  The report design of the pallet or container sublot label.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.

- **Range**: Any valid report design (version 1).

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre.

ProductionExecutionClient/PercentageOfVerticalScreenSize

- **Type**: BigDecimal
- **Value**: 0
- **Description**: Defines the vertical window size of the Production Execution Client.
  0: The default size is used.
  Any value <= 100: The specified percentage value of the current vertical screen size is used.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [0..100]

ProductionExecutionClient/ShowTickerPanel

- **Type**: Boolean
- **Value**: False
- **Description**: If the value is set to true, the ticker panel is shown in the status bar of the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Station/clearPasswordsDelay

- **Type**: Long
- **Value**: 3
- **Description**: The delay in seconds after which the phase completion signature panel loses the focus before entered passwords are cleared.
  A value of 0 means that there is no delay.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: >= 0

Station/useVirtualKeyboard

- **Type**: Boolean
- **Value**: True
- **Description**: Defines whether a virtual keyboard is displayed or not.
  If the value is set to **true**, the virtual keyboard is displayed.
  If the value is set to **false**, no virtual keyboard is displayed.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Configuration Keys Specific to EBR

The following configuration keys are available for EBR.

Activity/ExceptionViewInputActivity/WithRiskClassNone

- **Type**: Boolean
- **Value**: False
- **Description**: If the value is set to **true**, the **None** risk level is available when recording a user-defined exception in the Production Execution Client. This was default behavior up to PharmaSuite 8.0.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/apps-ebr-ifc.jar/CancelButtonEnabledOfExceptionDialog

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the **Cancel** button is available in the exception dialog of a system-triggered or user-triggered exception.
  If the value is set to **true**, the **Cancel** button is available thus allowing the user to discard the creation of an exception at that time.
  If the value is set to **false**, the **Cancel** button is not available thus forcing the user to create at least an **Exception canceled** exception.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/apps-ebr-ifc.jar/EBRServerRecoveryWorkerIntervalInMinutes

- **Type**: Long
- **Value**: 10
- **Description**: Defines the interval in minutes between runs of the EBR server recovery workers.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [1-60]

LibraryHolder/apps-ebr-ifc.jar/PhaseRepairModeBlackList

- **Type**: Object - List
- **Value**: PhaseRepairModeBlackList
- **Description**: Applies to the recovery capabilities of the Production Execution Client.
  The list contains the names (and optionally the revision) of phase system building blocks for which the repair capabilities are not available. If the revision is not included, all phase revisions are affected.
  Example: D Tare (RS) [5.2], D Tare (RS).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: A valid **List** object in Process Designer.

LibraryHolder/commons-messaging-ifc.jar/MessageTimeToLive

- **Type**: Long
- **Value**: 300000 (5 minutes)
- **Description**: The default time in milliseconds from its dispatch time that a produced message should be retained by the message system. The value of **0** means that there is no limit.
  For Automation Integration-specific settings see **MessageTimeToLive4AI** configuration key (page 85).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: >= 0

LibraryHolder/services-wip-impl.jar/DCSDefaultProducedMaterialLocation

- **Type**: Object - Location
- **Value**: N/A
- **Description**: The default storage location used for sublots produced by a DCS.

- **Evaluated**: When a produce sublot request is received from a DCS and the corresponding order step is not running. In case the order step is running, the storage location of the corresponding work center is used.
- **Range**: Any storage location.

MessageBrokerURL

- **Type**: String
- **Value**: tcp://<hostname>:<port>
- **Description**: Specifies the URL of the EBR server. Replace <hostname> by the name of the host to be used and <port> by the port number of the host to be used.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

MessagingTimeout

- **Type**: Long
- **Value**: 1000
- **Description**: The messaging timeout in milliseconds. The default value is 1000.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

PhaseUIExtensionsPrefix

- **Type**: String
- **Value**: PhaseUIExtensions
- **Description**: Prefix for the **UIExtensions** lists for phases.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Configuration Keys for Recipe, Order, and Workflow Management

The following configuration keys are available for recipe, order, and workflow management.

LibraryHolder/services-order-ifc.jar/allowNonValidMasterRecipeForOrder

- **Type**: Boolean
- **Value**: False

- **Description**: Allows to create orders for simulation master recipes (in a valid status or the **Verification** status) during order management and creates a separate control recipe with a complete copy of the underlying S88 structure during order explosion.
  If the value is set to **false**, the PharmaSuite **X_RefSelMasterRecipeFilterValidOnly** filter is applied, when assigning a master recipe to an order. Only master recipes in a valid status can be assigned. We recommend to use this configuration in a productive environment.
  If the value is set to **true**, the PharmaSuite **X_RefSelMasterRecipeFilterSimulation** filter is applied when assigning a master recipe to an order. Master recipes in a valid status or the **Verification** status can be assigned.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-order-ifc.jar/copyS88StructureFromMasterRecipeIntoControlRecipe

- **Type**: Boolean
- **Value**: False
- **Description**: Defines if S88 recipe entities of a master recipe are copied during order explosion and thus enables modifying the entities.
  If the value is set to **true**, the S88 recipe entities of a master recipe are copied and linked to the control recipe of its process order item. This configuration requires disk space.
  If the value is set to **false**, no control recipe-specific copy of S88 recipe entities are created.
- **Evaluated**: Each time an order is exploded.
- **Range**: [False, True]

LibraryHolder/services-order-ifc.jar/CostCenters

- **Type**: Object - List
- **Value**: CostCenters
- **Description**: Defines the name of the list of the cost centers available for shop floor-defined dispensing.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/services-order-ifc.jar/createSeparateControlRecipeFromValidMasterRecipe

- **Type**: Boolean
- **Value**: False
- **Description**: Creates a separate control recipe with a complete copy of the underlying S88 structure of a master recipe in a valid status during order explosion.
  If the value is set to **false**, no copy of a master recipe in a valid status and its S88 structure is created during order explosion.
  We recommend to use this configuration in a productive environment.
  If the value is set to **true**, a copy of the master recipe and its underlying S88 structure is created during order explosion (even if the master recipe is in a valid status).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-order-ifc.jar/DefaultOrderPrefix

- **Type**: String
- **Value**: SFD
- **Description**: Defines the default prefix of the automatically generated identifier of a shop floor-defined order (e.g. a cost center-related order).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Three characters.

LibraryHolder/services-order-ifc.jar/OrderNumberLength

- **Type**: Long
- **Value**: 12
- **Description**: Defines the default length of the automatically generated identifier of a shop floor-defined order (e.g. a cost center-related order).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [6..16]

LibraryHolder/services-recipe-impl.jar/useECBHazardSymbols

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, ECB hazard symbols are used. If the value is set to **false**, NFPA hazard symbols are used instead.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-recipe-impl.jar/useExtendedHazardSymbols

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, extended hazard symbols are used.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-recipe-impl.jar/useGHSData

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, GHS data (e.g. pictograms) is used.
  If the value is set to **false**, the **useECBHazardSymbols** configuration key (page 62) is evaluated. Additionally, you must
  - adapt the sublot labels in order to print ECB/NFPA-related data,
  - adapt the **Manage Materials** (ucRecipePart) user interface use case to display the ECB/NFPA-related tree node in the Production Management Client (see chapter "Using Forms in the Production Management Client" in Volume 1 of the "Technical Manual Configuration and Extension" [A7] (page 141)), and
  - adapt the *Part* (material) and *PartRevisionInfoModel* (audit trail) Data Dictionary classes in order to display the ECB/NFPA-related data in the Production Management Client (see chapter Adapting and Adding Field Attributes in Volume 2 of the "Technical Manual Configuration and Extension" [A3] (page 141)).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-s88-ifc.jar/BatchRecipeVersioningFSMName

- **Type**: String
- **Value**: MESMasterRecipeVersionGraph
- **Description**: Name of the versioning flexible state model used for batch-specific master recipes.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any versioning FSM configured for the **MasterRecipe** object in Process Designer.

LibraryHolder/services-s88-ifc.jar/DiscreteRecipeVersioningFSMName

- **Type**: String
- **Value**: MESMasterRecipeVersionGraph
- **Description**: Name of the versioning flexible state model used for device-specific master recipes.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any versioning FSM configured for the **MasterRecipe** object in Process Designer.

LibraryHolder/services-s88-ifc.jar/MasterWorkflowVersioningFSMName

- **Type**: String
- **Value**: MESMasterWorkflowVersionGraph
- **Description**: Name of the versioning flexible state model used for master workflows.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any versioning FSM configured for the **MasterRecipe** object in Process Designer. The **MasterRecipe** object is also used for master workflows.

LibraryHolder/services-S88-impl.jar/OneClickWorkflowOrderDefaultPrefix

- **Type**: String
- **Value**: WF
- **Description**: Defines the default prefix of the workflow order identifier for one-click workflows (see section Configuring Workflow Identifiers", chapter "Changing Number Generation Schemes" in Volume 2 of the "Technical Manual Configuration and Extension" ([A3] (page 141)).

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/services-S88-impl.jar/OneClickWorkflowOrderSequenceLength

- **Type**: Long
- **Value**: 8
- **Description**: Defines the length of the workflow order identifier for one-click workflows (see section "Configuring Workflow Identifiers", chapter "Changing Number Generation Schemes" in Volume 2 of the "Technical Manual Configuration and Extension" ([A3] (page 141)).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [8..15] (recommended)

LibraryHolder/services-wd-impl.jar/<usageType>.MasterRecipeQuantityCheckApplicable

- **Type**: Boolean
- **Value**: True
- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  If the value is set to **true**, the recipe structure-related *MasterRecipeQuantityCheck* check is executed (see **RecipeStructureChecks** configuration key (page 113)).
  If the value is set to **false**, the check is not executed unless material inputs or outputs have been assigned to the recipe.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-wd-impl.jar/<usageType>.MasterRecipeUsageTypeStructureCheckApplicable

- **Type**: Boolean
- **Value**: False

- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  If the value is set to **true**, the recipe structure-related *MasterRecipeUsageTypeStructureCheck* check is executed (see **RecipeStructureChecks** configuration key (page 113)).
  If the value is set to **false**, the check is not executed.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-wd-impl.jar/<usageType>.MaterialParameterCountsCheck Applicable

- **Type**: Boolean
- **Value**: True
- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  If the value is set to **true**, the recipe structure-related *MaterialParameterCountsCheck* check is executed (see **RecipeStructureChecks** configuration key (page 113)).
  If the value is set to **false**, the check is not executed.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-wd-impl.jar/CostCenter.MasterRecipeQuantityCheckApplicable

- **Type**: Boolean
- **Value**: False
- **Description**: Applies to the **Cost center** usage type:
  If the value is set to **true**, the recipe structure-related *MasterRecipeQuantityCheck* check is executed (see **RecipeStructureChecks** configuration key (page 113)).
  If the value is set to **false**, the check is not executed unless material inputs or outputs have been assigned to the recipe.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for

Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-wd-impl.jar/CostCenter.MasterRecipeUsageTypeStructureCheckApplicable

- **Type**: Boolean
- **Value**: True
- **Description**: Applies to the **Cost center** usage type:
  If the value is set to **true**, the recipe structure-related *MasterRecipeUsageTypeStructureCheck* check is executed (see **RecipeStructureChecks** configuration key (page 113)).
  If the value is set to **false**, the check is not executed.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-wd-impl.jar/CostCenter.MaterialParameterCountsCheckApplicable

- **Type**: Boolean
- **Value**: False
- **Description**: Applies to the **Cost center** usage type:
  If the value is set to **true**, the recipe structure-related *MaterialParameterCountsCheck* check is executed (see **RecipeStructureChecks** configuration key (page 113)).
  If the value is set to **false**, the check is not executed.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Configuration Keys for Inventory Management

The following configuration keys are available for inventory management.

LibraryHolder/services-discrete-impl.jar/DefaultDiscreteDeviceBarcodeTemplate

- **Type**: String
- **Value**: §ddddddddddddddddddddpppppppppppppppppp
- **Description**: Default template for device barcodes.
  The detail syntax is as follows:
  - **Value:** <prefix>
    - **Meaning**: Prefix of a barcode.
    - **Position in string**: Must be the first character, if present.
    - **Length**: 1
    - **Mandatory**: Yes, to allow scanning in the Cockpit and Start Processing views of the Production Execution Client
  - **Value**: <dddd...d>
    - **Meaning**: Device identifier. The number of 'd' characters determines the number of characters available for the device identifier (serial number). Leading and trailing spaces are cut off during processing.
    - **Position in string**: Arbitrary (apart from prefix, if present).
    - **Length**: Variable
    - **Mandatory**: Yes
  - **Value:** <pppp...p>
    - **Meaning**: Material identifier. The number of 'p' characters determines the number of characters available for the material identifier (name). Leading and trailing spaces are cut off during processing.
    - **Position in string**: Arbitrary (apart from prefix, if present).
    - **Length**: Variable
    - **Mandatory**: No
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/services-discrete-impl.jar/DiscreteDeviceBarcodeTemplates

- **Type**: List
- **Value**: (<value of **LibraryHolder/services-discrete-impl.jar/DefaultDiscreteDeviceBarcodeTemplate** configuration key>)
- **Description**: Device barcode templates. It must form a list of template strings, with each template string of the following form: **<prefix><dddd...d><pppp...p>.** For the exact syntax, see the description of the **LibraryHolder/services-discrete-impl.jar/DefaultDiscreteDeviceBarcodeTemplate** configuration key (page 71).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/services-inventory-impl.jar/<usageType>.BatchPrefix

- **Type**: String
- **Value**: BX
- **Description**: Applies to the specified usage type defined for shop floor-defined dispensing in the **RecipeUsageType** choice list:
  Defines the default prefix for batches.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Two characters.

> **TIP**
>
> When you modify this prefix, especially its length, you may influence the barcode-related configuration keys.

LibraryHolder/services-inventory-impl.jar/AreBatchIdentifiersUnique

- **Type**: Boolean
- **Value**: True
- **Description**: Defines whether pure batch identifiers (i.e. without material identifiers) are unique across PharmaSuite.
  If the value is set to **false**, you must adapt the following settings in addition to the configuration key itself:

- Clear the setting of the **Field length** parameter of the **batch** data dictionary element of the **OrderModel** data dictionary class. For this purpose, in Process Designer, run the **mes_DataDictManagerForm** form to start the **Data Dictionary Management** tool. Otherwise compound batch identifiers cannot be displayed in the **Order** form of the Production Management Client.
- Set the value of the**LibraryHolder/services-inventory-impl.jar/DefaultSublotBarcodeTemplate** configuration key (page 72) to $sssssssssbbbbbbbbbbpppppppppppppppppp in order to include the material identifier in the barcode.
- If the sublot label shall display the compound batch identifier, you must adapt the design of the sublot label. By default, the batch field of the label displays only the batch-related part of the batch identifier and the material identifier in a separate field. To change this behavior, in the report design use ((com.datasweep.compatibility.client.Batch)$P{P_BATCH}).getName() instead of $P{P_BATCH_IDENTIFIER} to determine the value of the batch field.
  Consider that additional layout changes may be required due to the length of the compound batch identifier.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

LibraryHolder/services-inventory-impl.jar/BatchBarcodeTemplates

- **Type**: List
- **Value**: (<value of **LibraryHolder/services-inventory-impl.jar/DefaultBatchBarcodeTemplate** configuration key>)
- **Description**: Batch barcode templates. It must form a list of template strings, with each template string of the following form: **<prefix><bbbb...b><pppp...p>.** For the exact syntax, see the description of the **LibraryHolder/services-inventory-impl.jar/DefaultBatchBarcodeTemplate** configuration key (page 71).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/services-inventory-impl.jar/BatchNumberLeadingZeros

- **Type**: Boolean
- **Value**: False
- **Description**: Specifies if the numeric part of a generated batch number has leading zeros.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

LibraryHolder/services-inventory-impl.jar/BatchNumberLength

- **Type**: Long
- **Value**: 10
- **Description**: Defines the maximum length of a batch number.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

> **TIP**
>
> When you modify this length, you may influence the barcode-related configuration keys.

LibraryHolder/services-inventory-impl.jar/CostCenter.BatchPrefix

- **Type**: String
- **Value**: CX
- **Description**: Applies to the **Cost center** usage type:
  Defines the default prefix for batches.
  See also section "Extension Use Case: Adding a Recipe-related Usage Type for Shop Floor-Defined Dispensing", chapter "Configuration and Extension" in "Technical Manual Phases of the Dispense Package" [A2] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Two characters.

> **TIP**
>
> When you modify this prefix, especially its length, you may influence the barcode-related configuration keys.

LibraryHolder/services-inventory-impl.jar/DefaultBatchBarcodeTemplate

- **Type**: String
- **Value**: #bbbbbbbbbbppppppppppp
- **Description**: Default template for batch barcodes.
  The detail syntax is as follows:
  - **Value:** <prefix>
    - **Meaning**: Prefix of a barcode.
    - **Position in string**: Must be the first character, if present.
    - **Length**: 1
    - **Mandatory**: Yes, to allow scanning in the Cockpit and Start Processing views of the Production Execution Client
  - **Value:** <bbbb...b>
    - **Meaning**: Batch identifier. The number of 'b' characters determines the number of characters available for the batch identifier (number). Leading and trailing spaces are cut off during processing.
    - **Position in string**: Arbitrary (apart from prefix, if present).
    - **Length**: Variable
    - **Mandatory**: Yes
  - **Value:** <pppp...p>
    - **Meaning**: Material identifier. The number of 'p' characters determines the number of characters available for the material identifier (name). Leading and trailing spaces are cut off during processing.
    - **Position in string**: Arbitrary (apart from prefix, if present).
    - **Length**: Variable
    - **Mandatory**: No
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/services-inventory-impl.jar/DefaultBatchPrefix

- **Type**: String
- **Value**: BX
- **Description**: Defines the default prefix for batches.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

> **TIP**
>
> When you modify this prefix, especially its length, you may influence the barcode-related configuration keys.

LibraryHolder/services-inventory-impl.jar/DefaultSublotBarcodeTemplate

- **Type**: String
- **Value**: $sssssssssbbbbbbbbbb
- **Description**: Default template for sublot barcodes.
  The detail syntax is as follows:
  - **Value:** <prefix>
    - **Meaning**: Prefix of a barcode.
    - **Position in string**: Must be the first character, if present.
    - **Length**: 1
    - **Mandatory**: No
  - **Value:** <ssss...s>
    - **Meaning**: Sublot identifier. The number of 's' characters determines the number of characters available for the sublot identifier. Leading and trailing spaces are cut off during processing.
    - **Position in string**: Arbitrary (apart from prefix, if present).
    - **Length**: Variable
    - **Mandatory**: Yes
  - **Value:** <bbbb...b>
    - **Meaning**: Batch identifier. The number of 'b' characters determines the number of characters available for the batch identifier (number). Leading and trailing spaces are cut off during processing.
    - **Position in string**: Arbitrary (apart from prefix, if present).
    - **Length**: Variable
    - **Mandatory**: Yes
  - **Value:** <pppp...p>

  - **Meaning**: Material identifier. The number of 'p' characters determines the number of characters available for the material identifier (name). Leading and trailing spaces are cut off during processing.
  - **Position in string**: Arbitrary (apart from prefix, if present).
  - **Length**: Variable
  - **Mandatory**: No
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

> **TIP**
>
> The default sublot barcode template is always a part of the **LibraryHolder/services-inventory-impl.jar/SublotBarcodeTemplates** configuration key (page 74). Otherwise you will not be able to read (parse) your own sublot labels.

LibraryHolder/services-inventory-impl.jar/DefaultSublotPrefix

- **Type**: String
- **Value**: SL
- **Description**: Defines the default prefix for sublot identifiers to be used when creating new sublots.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

LibraryHolder/services-inventory-impl.jar/IntraMaterialBatchPrefix

- **Type**: String
- **Value**: IX
- **Description**: Defines the prefix for batches assigned to an intra material.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

TIP

When you modify this prefix, especially its length, you may influence the barcode-related configuration keys.

LibraryHolder/services-inventory-impl.jar/isMaterialManagedInStock

- **Type**: Boolean
- **Value**: True
- **Description**: The value is set to **true**, if the system manages material in stock, otherwise the value is set to *false*.
  If the value is set to **false**, the following checks are never executed, regardless of whether any of them is configured somewhere else:
  - CheckBatchExpiryDate
  - CheckBOMSpecificMaterial
  - CheckMaterialForIntermediateOSI
  - CheckSublotBlockedByProcessing
  - CheckSublotProducedByOtherOS
  - CheckBatchQualityStatus
  - CheckSublotQuantity
  - CheckUndefinedBatchExpiryDate
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-inventory-impl.jar/SublotBarcodeTemplates

- **Type**: List
- **Value**: (<value of **LibraryHolder/services-inventory-impl.jar/DefaultSublotBarcodeTemplate** configuration key>)
- **Description**: Sublot barcode templates. It must form a list of template strings, with each template string of the following form: **<prefix><ssss...s><bbbb...b><pppp...p>**.
  For the exact syntax, see the description of the **LibraryHolder/services-inventory-impl.jar/DefaultSublotBarcodeTemplate** configuration key (page 72).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.

- **Range**: N/A

LibraryHolder/services-inventory-impl.jar/SublotIdentificationChecks

- **Type**: List
- **Value**: (CheckSublotNotContainedInLoadCarrier,
  CheckSublotQuantity,
  CheckBOMSpecificMaterial,
  CheckUndefinedBatchExpiryDate,
  CheckBatchExpiryDate,
  CheckBatchQualityStatus,
  CheckSublotBlockedByProcessing,
  CheckMaterialBatchSuggestedMaterialBatches,
  CheckMaterialForIntermediateOSI,
  CheckSublotProducedByOtherOS)
  - This list can be extended if new check classes are added during system configuration and extension.
- **Description**: A set of available sublot identification checks. A subset of these checks can be configured on the **OrderStepInput** level.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any element of the power set built from the allowed values.

Configuration Keys for Barcodes

The following configuration keys are available for barcodes.

Forms/2DBarcodeImages/characterSet

- **Type**: String
- **Value**: ISO-8859-1
- **Description**: Specifies the character set to be used to encode the barcode with the iText library of 2D barcode images on buttons. For the ZXing library, a character set cannot be defined.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Forms/2DBarcodeImages/encodingByIText

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the AGPL iText or Apache ZXing library is used to encode 2D barcode images on buttons.
  If the value is set to **true**, PharmaSuite uses iText.
  If the value is set to **false**, PharmaSuite uses ZXing.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LabelDesigns/2DBarcodeImages/characterSet

- **Type**: String
- **Value**: UTF-16
- **Description**: Specifies the character set to be used to encode the barcode with the iText library of 2D barcode images on labels and reports. For the ZXing library, a character set cannot be defined.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LabelDesigns/2DBarcodeImages/encodingByIText

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the AGPL iText or Apache ZXing library is used to encode 2D barcode images on labels and reports.
  If the value is set to **true**, PharmaSuite uses iText.
  If the value is set to **false**, PharmaSuite uses ZXing.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Configuration Keys for the Batch Report

The following configuration keys are available for the batch report.

LibraryHolder/services-batchreport-impl.jar/FontDirectory

- **Type**: String
- **Value**: %SystemRoot%/Fonts

- **Description**: The name of the directory containing the fonts to be used by PDF export. Usually *SystemRoot* is *c:/windows*.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid directory containing fonts.

LibraryHolder/services-batchreport-impl.jar/ImageNameOfBatchReportLogo

- **Type**: String
- **Value**: FTPS_Logo_BatchReport
- **Description**: The name of the FactoryTalk ProductionCentre image object used as logo within the batch production report, device history report, and workflow report.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid image name of an image present in the system.

LibraryHolder/services-batchreport-impl.jar/ImageNameOfMRReportLogo

- **Type**: String
- **Value**: FTPS_Logo_MRReport
- **Description**: The name of the FactoryTalk ProductionCentre image object used as logo within the master recipe report (batch, device) and the master workflow report.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid image name of an image present in the system.

LibraryHolder/services-batchreport-impl.jar/Main

- **Type**: Object - ReportDesign
- **Value**: PS-MRReport-Main
- **Description**: The report design of the master recipe report.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid report design (version 1).

TIP

Since PharmaSuite 7.0, the configuration key is no longer supported. It is replaced by the **LibraryHolder/services-batchreport-impl.jar/PS-MRReport-Main** configuration key (page 78).

LibraryHolder/services-batchreport-impl.jar/PS-BatchReport-EDHR-Main

- **Type**: Object - ReportDesign
- **Value**: PS-BatchReport-EDHR-Main
- **Description**: The report design of the electronic device history record.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid report design (version 1).

LibraryHolder/services-batchreport-impl.jar/PS-BatchReport-WorkflowOrder-Main

- **Type**: Object - ReportDesign
- **Value**: PS-BatchReport-WorkflowOrder-Main
- **Description**: The report design of the workflow order report.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid report design.

LibraryHolder/services-batchreport-impl.jar/PS-DRReport-Main

- **Type**: Object - ReportDesign
- **Value**: PS-DRReport-Main
- **Description**: The report design of the master recipe report (for device production).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid report design (version 1).

LibraryHolder/services-batchreport-impl.jar/PS-MRReport-Main

- **Type**: Object - ReportDesign
- **Value**: PS-MRReport-Main
- **Description**: The report design of the master recipe report (for batch production).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid report design (version 1).

LibraryHolder/services-batchreport-impl.jar/Workflow-Main

- **Type**: Object - ReportDesign
- **Value**: PS-MRReport-Workflow-Main
- **Description**: The report design of the workflow report.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid report design (version 1).

ReportPreview.useAlwaysPagination

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the report in the preview (Show report) includes pagination. The print preview always includes pagination.
  If the value is set to **true**, the preview includes pagination.
  If the value is set to **false**, the preview does not include pagination. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Configuration Keys for S88-specific Equipment

The following configuration keys are available for the S88-specific equipment processing.

Equipment/AIServerMessagingTimeoutInSeconds

- **Type**: Long
- **Value**: 60
- **Description**: The timeout for replies from the Automation Integration server for accessing FactoryTalk Live Data in seconds.
  Note: A FactoryTalk Live Data access itself may take more than 30 seconds in failure situations. However normal accesses only take milliseconds.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: We recommend to keep the default setting.
  Lowering the default setting may result in general timeout messages in case of FactoryTalk Live Data failure situations.

Equipment/DefaultAutomationIntegrationServerName

- **Type**: String
- **Value**: **AutomationIntegrationServer1**

- **Description**: Specifies the logical name of the default Automation Integration server.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Equipment/DefaultHistorianAccessServerName

- **Type**: String
- **Value**: HistorianAccessServer1
- **Description**: Specifies the logical name of the default Historian access server. If the provider is OSI PI, it is called **PI SQL Data Access Server**.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Equipment/DefaultHistorianAIServerName

- **Type**: String
- **Value**: AutomationIntegrationServer1
- **Description**: Specifies the logical name of the default Historian AI (Automation Integration) server.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Equipment/DefaultHistorianServerName

- **Type**: String
- **Value**: HistorianServer1
- **Description**: Specifies the logical name of the default Historian server. If the provider is OSI PI, it is called **PI-Server**.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Equipment/defaultLabelPrinter

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the default printer that is used for printing barcode labels for equipment entities.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid (local or network printer) name including double backslashes.

Equipment/DefaultLiveDataAreaPath

- **Type**: String
- **Value**: RNA://$Global/LiveDataArea
- **Description**: Path to the default FactoryTalk Live Data area (e.g. *RNA://@Global/...)*.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any valid path.

Equipment/DefaultTagUpdateRateInMilliseconds

- **Type**: Long
- **Value**: 2000
- **Description**: Update rate of tag values in milliseconds (msec).
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

Equipment/equipmentEntityBarcode.prefix

- **Type**: String
- **Value**: N/A
- **Description**: Defines the prefix of the barcode label for equipment entities.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Equipment/equipmentEntityLabel.reportDesign

- **Type**: Object - ReportDesign
- **Value**: s88EquipmentEntityLabelRD
- **Description**: The report design of the barcode label for equipment entities.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid report design.

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre.

Equipment/equipmentEntityLabel.showPrinterSelection

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, a print dialog box is displayed before printing.
  If the value is set to **false**, no print dialog box is displayed before printing.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Equipment/equipmentEntityLabel.subroutine

- **Type**: Object - Subroutine
- **Value**: s88EquipmentEntityLabelSR
- **Description**: The subroutine to retrieve the data required according to the report design for equipment entities.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid subroutine that supports all used equipment entity label report designs.

Equipment/equipmentEntityLabel.usePreviewWindowForTouchscreen

- **Type**: Boolean
- **Value**: False
- **Description**: If the value is set to **true**, Data Manager uses the preview window for touch screens (used in Production Execution Client).
  If the value is set to **false**, Data Manager uses the standard preview window (also used in Production Execution Client or Recipe and Workflow Designer).

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Equipment/HistorianAccessPassword.<HistorianAccessServerName>

- **Type**: String
- **Value**: N/A
- **Description**: Encrypted password to log in to the specified Historian access server.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Equipment/HistorianAccessUser.<HistorianAccessServerName>

- **Type**: String
- **Value**: N/A
- **Description**: User name to log in to the specified Historian access server.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Equipment/LiveDataPassword

- **Type**: String
- **Value**: null (the logged-in Windows user is used)
- **Description**: Encrypted or unencrypted password to connect to FactoryTalk Live Data.
  This is also used by the Automation Integration server.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Password of a user configured in FactoryTalk Live Data.

Equipment/LiveDataUsername

- **Type**: String
- **Value**: null (the logged-in Windows user is used)
- **Description**: User name to connect to FactoryTalk Live Data.
  This is also used by the Automation Integration server.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Name of a user configured in FactoryTalk Live Data.

Equipment/PI_JDBC_DCA_OPTION

- **Type**: String
- **Value**: INIT
- **Description**: Applies only if the provider is OSI PI.
  Defines the behavior (ignore or reuse) of persisted authentication (*filename.dca*) for PI SQL DAS:
  - INIT: clears persisted authentication information,
  - REUSE: reuses persisted authentication,
  - file name: uses specific authentication file,
  - SAVE: persists a DCA file from values specified in user and password properties.

  For details, see OSIsoft PI JDBC 2010 R3 Administrator Guide, [D2] (page 142).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Equipment/PI_JDBC_Property_LogConsole

- **Type**: Boolean
- **Value**: False
- **Description**: Applies only if the provider is OSI PI.
  Specifies if log messages of the PI JDBC driver are printed to the console. For details, see OSIsoft PI JDBC 2010 R3 Administrator Guide, [D2] (page 142).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Equipment/PI_JDBC_Property_LogFile

- **Type**: String
- **Value**: N/A
- **Description**: Applies only if the provider is OSI PI.
  Specifies the full log file path/name of the PI JDBC driver. For details, see OSIsoft PI JDBC 2010 R3 Administrator Guide, [D2] (page 142).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Equipment/PI_JDBC_Property_LogLevel

- **Type**: Long
- **Value**: 0
- **Description**: Applies only if the provider is OSI PI.
  Specifies the granularity of messages to be logged by the PI JDBC driver. For details, see OSIsoft PI JDBC 2010 R3 Administrator Guide, [D2] (page 142).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [0..5]

Equipment/PI_JDBC_URL

- **Type**: String
- **Value**: jdbc:pisql://{0}/Data Source={1}; Integrated Security=SSPI
- **Description**: Applies only if the provider is OSI PI.
  Specifies the URL to connect the PI JDBC driver, where the first parameter is the name of the PI DA server and the second parameter is the name of the PI server. For details, see OSIsoft PI JDBC 2010 R3 Administrator Guide, [D2] (page 142).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Equipment/TagQualityGoodList

- **Type**: List
- **Value**: TagGoodQuality
- **Description**: List holding all numeric values that are considered good tag quality.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Values of **FTLDConstants.OPCQuality*** constants.

LibraryHolder/commons-messaging-ifc.jar/MessageTimeToLive4AI

- **Type**: Long
- **Value**: 3000 (3 seconds)
- **Description**: For Automation Integration, the default time in milliseconds from its dispatch time that a produced message should be retained by the message system. The value of **0** means that there is no limit.
- **Evaluated**: Only when a PharmaSuite client is started.

- **Range**: >= 0

LibraryHolder/services-eqm-impl.jar/RoomCleaningRules

- **Type**: List
- **Value**: (CleaningRuleInputMaterialHasChanged, CleaningRulePredecessorWasActiveSubstance, CleaningRulePredecessorMaterialUnknown)
  - This list can be extended if new check classes are added during system configuration and extension.
- **Description**: A set of rules (checks) to determine the cleaning demand of an intended binding for the current room.
  By default, the following checks are available:
  - CleaningRuleInputMaterialHasChanged: Checks whether the input material to be dispensed has been changed.
  - CleaningRulePredecessorWasActiveSubstance: Checks whether the input material to be dispensed has been changed and the previous input material was an active substance.
  - CleaningRulePredecessorMaterialUnknown: Checks whether the input material is known and the predecessor material is unknown or does not exist at all.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/services-s88eqmreport-impl.jar/ImageNameOfEQMReportLogo

- **Type**: String
- **Value**: FTPS_Logo
- **Description**: The name of the FactoryTalk ProductionCentre image object used as logo within the equipment entity logbook report.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid image name of an image present in the system.

LibraryHolder/services-s88equipment-ifc.jar/FlexibleStateModelS88EquipmentClass

- **Type**: String
- **Value**: S88EquipmentClassStatusGraph
- **Description**: Name of the flexible state model used for the status management of S88 equipment classes.

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for an **FSM** object in Process Designer.

LibraryHolder/services-s88equipment-ifc.jar/FlexibleStateModelS88EquipmentEntity

- **Type**: String
- **Value**: S88EquipmentEntityStatusGraph
- **Description**: Name of the flexible state model used for the status management of S88 equipment entities.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for an **FSM** object in Process Designer.

LibraryHolder/services-s88equipment-ifc.jar/FlexibleStateModelS88StatusGraph

- **Type**: String
- **Value**: S88StatusGraphChangeControl
- **Description**: Name of the flexible state model used for the status management of S88 equipment status graphs.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for an **FSM** object in Process Designer.

LibraryHolder/services-s88equipment-impl.jar/GeneratedEntityBarcodePrefix

- **Type**: String
- **Value**: "" (i.e. empty string)
- **Description**: Defines the default barcode prefix for equipment entities generated from a template entity during processing.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any character or none (i.e. empty string)

LibraryHolder/services-s88equipment-impl.jar/GeneratedEntityIdentifierPrefix

- **Type**: String
- **Value**: EQ
- **Description**: Defines the default identifier prefix of the automatically generated identifier for equipment entities generated from a template entity during processing.
- **Evaluated**: Only when a PharmaSuite client is started.

- **Range**: N/A
  The prefix should be shorter than the length defined with the **LibraryHolder/services-s88equipment-impl.jar/GeneratedEntityLengthOfNumericPartOfIdentifier** configuration key (page 88).

LibraryHolder/services-s88equipment-impl.jar/GeneratedEntityLengthOfNumericPartOfIdentifier

- **Type**: Long
- **Value**: 10
- **Description**: Defines the default length of the numeric part of the automatically generated identifier for equipment entities generated from a template entity during processing.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [5..80]
  The minimum depends on the **LibraryHolder/services-s88equipment-impl.jar/GeneratedEntityIdentifierPrefix** configuration key (page 87).

services-s88equipment-ifc.jar/FlexibleStateModelBinding

- **Type**: String
- **Value**: S88EquipmentBinding
- **Description**: Name of the flexible state model used for S88 equipment binding.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Name of the flexible state model assigned to the **binding** FSM relationship.

Station/defaultLabelPrinter

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the default printer that is used for printing barcode labels for stations.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid (local or network printer) name including double backslashes.

Station/stationBarcode.prefix

- **Type**: String
- **Value**: &

- **Description**: Defines the prefix of the barcode label for stations.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Station/stationLabel.reportDesign

- **Type**: Object - ReportDesign
- **Value**: MESStationLabelRD
- **Description**: The report design of the barcode label for stations.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid report design.

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre.

Station/stationLabel.showPrinterSelection

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, a print dialog box is displayed before printing.
  If the value is set to **false**, no print dialog box is displayed before printing.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Station/stationLabel.subroutine

- **Type**: Object - Subroutine
- **Value**: MESStationLabelSR
- **Description**: The subroutine to retrieve the data required according to the report design for stations.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid subroutine.

Station/stationLabel.usePreviewWindowForTouchscreen

- **Type**: Boolean
- **Value**: False

- **Description**: If the value is set to **true**, Data Manager uses the standard preview window (also used in Production Execution Client or Recipe and Workflow Designer).
  If the value is set to **false**, Data Manager uses the preview window for touch screens (used in Production Execution Client).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Configuration Keys for Exception Recording

The following configuration keys are available for exception recording.

Libraries/apps-ebr-impl.jar/ContainerStatusFailureExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: When a source sublot is identified that is associated with a container entity, the phase checks if the approval status of the container is **Approved**. A **Container status check** system-triggered exception is displayed automatically if the check has failed. The configuration key defines the exception-specific risk level of the exception.
  Applies only to the **D Identify material (RS) [5.0] (MR1)** phase.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: The meaning of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-ifc.jar/PhaseExecutionSkippedExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: Applies to the recovery capabilities within the Production Execution Client.
  The exception-specific risk level of a system-triggered exception added to an active phase, when aborting the phase is confirmed.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-ifc.jar/PhaseRepairModeExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: Applies to the recovery capabilities within the Production Execution Client.
  The exception-specific risk level of a system-triggered exception added to an active phase, when the repair mode of a phase is started or when process parameters have been edited in the repair mode of a phase.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-impl.jar/AutomaticExceptionOnResumeOfServerRunOperationExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an automatic exception added to all active operations that have been resumed, when the Operation Execution (OE) server is restarted.
- **Evaluated**: Only when the Operation Execution (OE) server is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-impl.jar/CanceledExceptionRisk

- **Type**: Long
- **Value**: 10
  (Value of the **Low risk** level)
- **Description**: The exception-specific risk level of the "Exception canceled" exception.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-impl.jar/HoldOrResumeOESOperationExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an exception added to the active runtime operation, when the operation is held or resumed with the PharmaSuite Administrator Console. For more information, see chapter "Monitoring PharmaSuite and Related Components" in "Technical Manual Administration" [A6] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-impl.jar/OperationUnlockedByAdminExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an exception added to all active phases of an operation, when the operation is unlocked and unregistered with the **Remove Object Locks** task in the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-impl.jar/OSCanceledWhilePhaseActiveExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of a system-triggered exception added to all active phases when an order or workflow is canceled or a unit procedure is aborted.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-impl.jar/RoomTriggerStatusTransitionFailureExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)

- **Description**: The exception-specific risk level of a system-triggered exception in case a room-specific status transition fails. Applies to system phase building blocks of the Dispense package that do not hold the **Status transition failed** process parameter.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-impl.jar/SequentialPhaseCompletionUserAlreadySignedExcRisk

- **Type**: Long
- **Value**: 35
  (Value of the **High risk with mandatory comment** level)
- **Description**: The exception-specific risk level of an exception added to a phase if a sequential completion signature is enabled and a user who has already signed for a previous phase completion within this operation signs for the phase completion.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-ebr-impl.jar/UnloadOrderFromEBRExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of the exceptions added to the active runtime unit procedures when an order (workflow) is unloaded or loaded with the PharmaSuite Administrator Console. For more information, see chapter "Monitoring PharmaSuite and Related Components" in "Technical Manual Administration" [A6] (page 141).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/apps-masterdata-impl.jar/ForceUnbindEquipmentExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)

- **Description**: The exception-specific risk level of an automatic exception added to the runtime unit procedure of a bound entity, when the **Force unbinding of equipment entity** function in Data Manager has been performed.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/commons-deviation-impl.jar/ExceptionSequencerLength

- **Type**: Long
- **Value**: 9
- **Description**: The minimum length of generic exception record identifiers.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any value greater than 3, since **EXC** is the common prefix used by exception records created through this service.

LibraryHolder/services-eto-impl.jar/AutomaticExceptionOnFailureAtCreationOfRunOfETOExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an automatic exception added to an event-triggering phase, when the creation of an ETO run fails e.g. because the Triggered Operation Management (TOM) server does not run.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-eto-impl.jar/AutomaticExceptionOnResumeOfTemplateExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an automatic exception added to all active event-triggered operations (templates) that have been resumed, when the Triggered Operation Management (TOM) server is restarted.
- **Evaluated**: Only when the Triggered Operation Management (TOM) server is started.

- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-order-ifc.jar/ForceTransitionExecutionOrderExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an exception added to the corresponding runtime entity (runtime unit procedure, operation, or phase), when a transition is forced for orders (workflows) in the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-order-ifc.jar/UnitProcedureAbortByAdminExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an exception added to the active runtime unit procedure, when the corresponding non-Dispense order step has been aborted in the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-order-ifc.jar/UnitProcedureFinishWithAbortedByAdminExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an exception added to the aborted runtime unit procedures, when the corresponding order has been finished in the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-order-ifc.jar/UnitProcedureReactivatedByAdminExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an exception added to the active runtime unit procedure, when the corresponding non-Dispense order step has been reactivated in the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-s88equipment-impl.jar/SaveReadOnlyEquipmentExcRisk

- **Type**: String
- **Value**: High
  (Meaning of the **RiskClass** choice list)
- **Description**: The exception-specific risk level of an exception added to the runtime unit procedure of a bound entity (in a read-only status, e.g. **Verification**, **Approved**), when runtime data, such as runtime properties, graph-related data (e.g. status, expiry date), and grouping situations, of the equipment entity was changed in Data Manager.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The meaning of the elements of the **RiskClass** choice list.

LibraryHolder/services-S88-ifc.jar/ExternalExceptionHandlingReceiver

- **Type**: String
- **Value**: QMS
- **Description**: Defines the logical name of an external system that receives exceptions from a PharmaSuite system and responds to them.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/services-S88-ifc.jar/ExternalExceptionHandlingSender

- **Type**: String
- **Value**: MES
- **Description**: Defines the logical name of a PharmaSuite system that sends exceptions to an external system.
- **Evaluated**: Only when a PharmaSuite client is started.

- **Range**: N/A

LibraryHolder/services-S88-impl.jar/ActivateExternalExceptionHandling

- **Type**: Boolean
- **Value**: False
- **Description**: Defines if an exception is sent to an external QMS for review in case a defined criterion (e.g. minimal risk level) is fulfilled.
  If the value is set to **true**, exceptions are sent to an external QMS. Additionally, the **Communication status** and the **External ID** are shown in the UI.
  If the value is set to **false**, exceptions are not sent to an external system. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: **Range**: [False, True]

LibraryHolder/services-S88-impl.jar/AppendWorkflowsToOrderInPECExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an exception added to the active runtime unit procedure, when orders (workflows) are appended to the corresponding order step in the Production Execution Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-S88-impl.jar/AppendWorkflowsToOrderInPECSignatureEnabled

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether an exception (along with an electronic signature) is recorded when orders (workflows) are appended to the corresponding order step in the Production Execution Client.
  If the value is set to **true**, the Production Execution Client requests to record an electronic signature.
  If the value is set to **false**, orders (workflows) can be appended without signing for the action. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: **Range**: [False, True]

LibraryHolder/services-S88-impl.jar/CommunicationStatesToAllowManualExceptionSending

- **Type**: String - List
- **Value**: Not sent, Error (meanings of the *ExceptionCommunicationStatus* choice list)
- **Description**: Specifies the possible communication states for which it is allowed to send an exception to an external QMS. The states are defined as meanings of the corresponding choice elements of the *ExceptionCommunicationStatus* choice list.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the meaning of the *ExceptionCommunicationStatus* choice list.

LibraryHolder/services-S88-impl.jar/MinimalRiskClassForExternalExceptionHandling

- **Type**: String
- **Value**: High
  (Meaning of the **RiskClass** choice list)
- **Description**: The minimum risk class of an exception to be handled externally.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-S88-impl.jar/OrdersAssignedByAdminExcRisk

- **Type**: Long
- **Value**: 30
  (Value of the **High risk** level)
- **Description**: The exception-specific risk level of an exception added to the active runtime unit procedure, when the set of orders (workflows) appended to the corresponding order step has been changed in the Production Management Client.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: The values of the elements of the **RiskClass** choice list.

LibraryHolder/services-s88-impl.jar/S88ExceptionSequencerLength

- **Type**: Long
- **Value**: 3
- **Description**: The minimum length of exception record identifiers.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: Any value greater than 0.

Configuration Keys for Dispense

The following configuration keys are available for Dispense-related processes.

LabelDesign/defaultPrinter

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the default printer that is used for printing labels.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: A valid (local or network printer) name including double backslashes.

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LabelDesign/dispensingLabel.print.printLabelWhenTargetSublotCreated

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether a target sublot label will be printed when a target sublot is created.
  If the value is set to **true**, a target sublot label will be printed when a target sublot is created.
  If the value is set to **false**, no target sublot label will be printed when a target sublot is created.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

TIP

This setting can be combined with the **LabelDesign/dispensingLabel.print.printLabelWhenTargetSublotFinished** configuration key (page 100).

In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LabelDesign/dispensingLabel.print.printLabelWhenTargetSublotFinished

- **Type**: Boolean
- **Value**: True
- **Description**: Defines whether a target sublot label will be printed when a target sublot is finished.
  If the value is set to **true**, a target sublot label will be printed when a target sublot is finished.
  If the value is set to **false**, no target sublot label will be printed when a target sublot is finished.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

TIP

This setting can be combined with the **LabelDesign/dispensingLabel.print.printLabelWhenTargetSublotCreated** configuration key (page 99).

In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LabelDesign/dispensingLabel.reprint.showPreview

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, label reprints will be opened in preview mode.
  If the value is set to **false**, label reprints will be printed without preview.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

TIP

In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LabelDesign/dispensingLabel.reprint.showPrinterSelection

- **Type**: Boolean
- **Value**: False
- **Description**: If the value is set to **true**, a print dialog box will be displayed before reprinting.
  If the value is set to **false**, no print dialog box will be displayed before reprinting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

TIP

In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LabelDesign/inventoryLabel.print.printInvLabelWhenSourceSublotConsumed

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether a source sublot label will be printed when a target sublot is consumed.
  If the value is set to **true**, a source sublot label will be printed when a target sublot is consumed.
  If the value is set to **false**, no source sublot label will be printed when a target sublot is consumed.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [False, True]

TIP

In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LabelDesign/jasperReportViewerPanelClass

- **Type**: String
- **Value**: com.rockwell.mes.commons.base.ifc.labeling.MESJasperReportViewerPanel
- **Description**: Defines the class, in this a case a panel, to be used in order to show the preview of a label.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: A fully qualified class name of a subclass of *MESJasperReportViewerPanel* or the *MESJasperReportViewerPanel* class itself.

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LabelDesign/simpleWeighingLabel.reprint.showPreview

- **Type**: Boolean
- **Value**: True
- **Description**: If the value is set to **true**, label reprints will be opened in preview mode.
  If the value is set to **false**, label reprints will be printed without preview.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LabelDesign/simpleWeighingLabel.reprint.showPrinterSelection

- **Type**: Boolean
- **Value**: False
- **Description**: If the value is set to **true**, a print dialog box will be displayed before reprinting.
  If the value is set to **false**, no print dialog box will be displayed before reprinting.
- **Evaluated**: Only when a PharmaSuite client is started.

- **Range**: [False, True]

> **TIP**
>
> In PharmaSuite, labels are based on the **ReportDesign** object of FactoryTalk ProductionCentre, although a **LabelDesign** prefix is used.

LibraryHolder/apps-wd-impl.jar/BatchQualityFSMName

- **Type**: String
- **Value**: BatchQuality
- **Description**: Defines the FSM name configured for the batch-related **BatchQuality** FSM relationship. The default value is **BatchQuality**.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid FSM name.

LibraryHolder/services-wd-impl.jar/OrderStepInputComparisonPattern

- **Type**: String
- **Value**: 0000
- **Description**: A **DecimalFormat** pattern to format the position and the split number during the **OrderStepInput** comparison.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: N/A

ReportDesign/defaultPrinter

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the default printer that is used for printing reports.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: A valid (local or network printer) name including double backslashes.

Station/GrossWeighing.weighMode

- **Type**: Long
- **Value**: 0
- **Description**: Defines the behavior of the scales control, when displaying weighing values. It can be defined separately for each weighing method (net weighing, gross weighing, and removal weighing).
  If the value is set to **0**, the weighing value is displayed from zero, and the starting point is zero.
  If the value is set to **1**, the weighing value is displayed against zero, and the starting point is the negative nominal quantity.
  Values other than 0 and 1 are not specified.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: 0, 1

Station/NetWeighing.weighMode

- **Type**: Long
- **Value**: 0
- **Description**: Defines the behavior of the scales control, when displaying weighing values. It can be defined separately for each weighing method (net weighing, gross weighing, and removal weighing).
  If the value is set to **0**, the weighing value is displayed from zero, and the starting point is zero.
  If the value is set to **1**, the weighing value is displayed against zero, and the starting point is the negative nominal quantity.
  Values other than 0 and 1 are not specified.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: 0, 1

Station/RemovalWeighing.weighMode

- **Type**: Long
- **Value**: 0
- **Description**: Defines the behavior of the scales control, when displaying weighing values. It can be defined separately for each weighing method (net weighing, gross weighing, and removal weighing).
  If the value is set to **0**, the weighing value is displayed from zero, and the starting

point is zero.
If the value is set to **1**, the weighing value is displayed against zero, and the starting point is the negative nominal quantity.
Values other than 0 and 1 are not specified.

- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: 0, 1

Configuration Keys for Recipe and Workflow Designer

The following configuration keys are available for Recipe and Workflow Designer.

AccessPrivilege/ConfidentialObject.Modifiable.WhenCreatingRecipeStructure

- **Type**: Boolean
- **Value**: False
- **Description**: Applies only if the PharmaSuite system is configured for confidential objects.
  If an access privilege is defined for a material or an ERP BOM and a new master recipe, master workflow, or building block is created, the access privilege can be taken over to the new master recipe, master workflow, or building block.
  If the value is set to **true**, the access privilege taken over from the material or ERP BOM can be changed.
  If the value is set to **false**, the access privilege taken over from the material or ERP BOM cannot be changed. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/ApplicationStart_RecipeDesigner/StartupChecks

- **Type**: List
- **Value**: N/A
- **Description**: List of configurable checks to be executed when Recipe and Workflow Designer is started.
  Each check must implement the *IStartupCheck* interface.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for a **List** object in Process Designer.

Form/ApplicationStart_RecipeDesigner/StyleSheet

- **Type**: Object - List
- **Value**: red_StyleSheet
- **Description**: Defines the style sheet used in Recipe and Workflow Designer.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/RecipeDesigner/ps_ExpressionEditorShortcuts.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the shortcuts of the Expression editor.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/RecipeDesigner/ps_ExpressionEditorShortcuts.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration of the shortcuts of the Expression editor is read from a file.
  If the value is set to **true**, Recipe and Workflow Designer reads the XML configuration from the *ps_ExpressionEditorShortcuts.xml* file. This configuration is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
  If the value is set to **false**, Recipe and Workflow Designer reads the XML configuration from the database (**ps_ExpressionEditorShortcuts** list).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/RecipeDesigner/red_buttonConfiguration.customerName

- **Type**: String
- **Value**: N/A

- **Description**: Defines the name of the XML configuration that is used to configure the menu bar, toolbar, and context menus.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/RecipeDesigner/red_buttonConfiguration.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration of the menu bar, toolbar, and context menus is read from a file.
  If the value is set to **true**, Recipe and Workflow Designer reads the XML configuration from the *red_buttonConfiguration.xml* file. This configuration is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
  If the value is set to **false**, Recipe and Workflow Designer reads the XML configuration from the database (**red_buttonConfiguration** list).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/RecipeDesigner/red_defaultLayoutSettings.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the UI configuration of the floating windows.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/RecipeDesigner/red_defaultLayoutSettings.readFromFile

- **Type**: Boolean
- **Value**: False

- **Description**: Defines whether the UI configuration of the floating windows is read from a file.
  If the value is set to **true**, Recipe and Workflow Designer reads the configuration from the *red_defaultLayoutSetting.ini* file located in the user-specific directory. This configuration is recommended only for developers.
  If the value is set to **false**, Recipe and Workflow Designer reads the configuration from the database (**red_defaultLayoutSetting** list).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/RecipeDesigner/red_expressionContextFunctionConfiguration.customerName

- **Type**: String
- **Value**: red_expressionContextFunctionConfiguration
- **Description**: Defines the name of the XML configuration that is used to configure the context-related functions of the Expression editor.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/RecipeDesigner/red_ExpressionEditor.showErrorIconMargin

- **Type**: Boolean
- **Value**: True
- **Description**: Defines whether the Expression editor shows error icons.
  If the value is set to **true**, the Expression editor shows an error icon to the left of each affected line.
  If the value is set to **false**, no error icons are shown.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/RecipeDesigner/red_ExpressionEditor.showLineNumbers

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the Expression editor shows line numbers.
  If the value is set to **true**, the Expression editor shows a line number to the left of each line.
  If the value is set to **false**, no line numbers are shown.

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/RecipeDesigner/red_materialConfigurationDiscrete.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the **Material** dialog for Recipe Designer - Device.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/Universe/red_materialConfiguration.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the **Material** dialog for Recipe Designer - Batch.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/Universe/red_materialConfiguration.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration is read from a file.
  If the value is set to **true**, Recipe and Workflow Designer reads the XML configuration of the **Material** dialog from the *red_materialConfiguration.xml* file. This configuration slows down the system and is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
  If the value is set to **false**, Recipe and Workflow Designer reads the XML configuration from the database (**red_materialConfiguration** list).
- **Evaluated**: Only when a PharmaSuite client is started.

- **Range**: [False, True]

Form/Universe/red_materialConfiguration.refreshAlways

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration is updated.
  If the value is set to **true**, the **Material** dialog re-reads the XML configuration when the user clicks the **Refresh** button. This configuration slows down the system and is recommended only for developers. For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/Universe/red_openConfiguration.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the **Open** dialog.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/Universe/red_openConfiguration.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration is read from a file.
  If the value is set to **true**, Recipe and Workflow Designer reads the XML configuration of the **Open** dialog from the *red_openConfiguration.xml* file. This configuration slows down the system and is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
  If the value is set to **false**, Recipe and Workflow Designer reads the XML configuration from the database (**red_openConfiguration** list).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/Universe/red_openConfiguration.refreshAlways

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration is updated.
  If the value is set to **true**, the **Open** dialog re-reads the XML configuration when the user clicks the **Refresh** button. This configuration slows down the system and is recommended only for developers. For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/Universe/red_universeConfiguration.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the Universe window.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/Universe/red_universeConfiguration.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration is read from a file.
  If the value is set to **true**, Recipe and Workflow Designer reads the XML configuration of the Universe window from the *red_universeConfiguration.xml* file. This configuration slows down the system and is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
  If the value is set to **false**, Recipe and Workflow Designer reads the XML configuration from the database (**red_universeConfiguration** list).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/Universe/red_universeConfiguration.refreshAlways

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration is updated.
  If the value is set to **true**, the Universe window re-reads the XML configuration when the user clicks the **Refresh** button. This configuration slows down the system and is recommended only for developers. For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/apps-recipeeditor-impl.jar/ImportRecipeBuildingBlockIgnoreNonExistingEquipment

- **Type**: Boolean
- **Value**: False
- **Description**: Determines the result of the import of master recipes, master workflows, or building blocks and how work center assignment parameters are handled if the referenced work center or station object does not exist in the target system.
  If the value is set to **true**, the work center assignment parameters will not be added. The import is completed with warnings. Under specific circumstance, it may become necessary to allow this behavior. We highly recommend to use this configuration with care, since it might result in differences between the exported object and the imported object (work center assignment parameters, status of the object). For more information, see chapter "Exporting and Importing Master Recipes, Master Workflows, and Building Blocks" in "Technical Manual Administration" [A6] (page 141).
  If the value is set to **false**, the import fails. We recommend to use this configuration in a productive environment.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/apps-recipeeditor-impl.jar/InlineWeighingPhaseLibs

- **Type**: Object - List
- **Value**: InlineWeighingPhaseLibs
- **Description**: The list contains the name (and revision) of Inline Weighing-specific system building blocks for which the Inline Weighing-specific recipe structure checks are evaluated if a material parameter is specified for a

phase in this list. Additionally, the unit procedure and operation must not be Dispense-specific.
Example: D Identify Material (RS) [3.0], D Identify Material (RS) [5.0]

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid **List** object in Process Designer.

LibraryHolder/apps-recipeeditor-impl.jar/OnlyOneMFCInputPerDispenseConfluence

- **Type**: Boolean
- **Value**: True
- **Description**: Determines if a confluence in a Dispense unit procedure may have only one or more than one incoming link.
  If the value is set to **true**, a confluence in a Dispense unit procedure may have only one incoming link (non-layered dispensing).
  If the value is set to **false**, a confluence in a Dispense unit procedure may have more than one incoming link (layered dispensing).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/apps-recipeeditor-impl.jar/OutputWeighingPhaseLibs

- **Type**: Object - List
- **Value**: OutputWeighingPhaseLibs
- **Description**: The list contains the name (and revision) of Output Weighing-specific system building blocks for which the Output Weighing-specific recipe structure checks are evaluated if a material parameter is specified for a phase in this list.
  Example: O Weigh (RS) [1.0], O Weigh (RS) [5.0]
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid **List** object in Process Designer.

LibraryHolder/apps-recipeeditor-impl.jar/RecipeStructureChecks

- **Type**: Object - List
- **Value**: (AutostartableCapabilityNotInSimultaneousBranchCheck
  BOMPositionNotEmptyCheck
  CapabilityCombinationCheck
  CapabilityEscalationDurationsCheck
  CapabilityRisksMatchedByPrivilegeUsagesCheck
  CapabilityTriggerableHasChildCheck

ChildrenCapabilititesCombinationCheck
DispenseFlagCapabilityParameterCheck
EquipmentPropertyRequirementsCheck
EquipmentPropertyRequirementIdentifierCheck
EquipmentRequirementParameterCountsCheck
EquipmentRequirementParameterUniqueBOEPositionCheck
EquipmentRequirementUniqueClassCheck
EquipmentRuleExpressionCheck
ERPBomAllPositionsUsedInActualBomCheck
ERPBomContainsAllActualBomPositionsCheck
ERPBomItemConsistentMaterialParameterAttributesCheck
ERPBomPackagingLevelsCheck
ERPBomQuantityCheck
ERPBomUniqueMFCPositionCheck
InputExpressionsCheck
MasterRecipeMinMaxQuantityCheck
MasterRecipePackagingInventoryLevelCheck
MasterRecipePackagingLevelsConsistencyCheck
MasterRecipeQuantityCheck
MasterRecipeUsageTypeStructureCheck
MaterialParameterActivePlannedPotencyForInlineCheck
MaterialParameterAllowedWeighingMaterialTypeForInlineWeighingCheck
MaterialParameterAllowedWeighingMethodsForInlineWeighingCheck
MaterialParameterAsDefinedPlannedQuantityAndTolerancesForInlineWeighingCheck
MaterialParameterAsDefinedPlannedQuantityAndTolerancesForOutputWeighingCheck
MaterialParameterAsProducedRequiresTolerancesCheck
MaterialParameterCountsCheck
MaterialParameterDispenseCheck
MaterialParameterEmptyMFCPositionForInlineWeighingCheck
MaterialParameterFixedQuantityCheck
MaterialParameterNumericMFCPositionCheck
MaterialParameterPackagingInventoryLevelCheck
MaterialParameterPackagingLevelsCheck
MaterialParameterPackagingLevelsConsistencyCheck
MaterialParameterPlannedPotencyUomCheck
MaterialParameterPlannedQuantityCheck
MaterialParameterPlannedQuantityModeAsProducedCheck
MaterialParameterPlannedQuantityModeForDispenseCheck
MaterialParameterPlannedQuantityModeNoneCheck
MaterialParameterPlannedQuantityModeOfTransferInForInlineWeighingCheck
MaterialParameterQuantityUomCheck
MaterialParametersWithSameMFCPositionConsistencyCheck

MaterialParameterTargetWeightQuantityAndTolerancesCheck
MaterialParameterTargetWeightQuantityUomCheck
MaterialParameterTargetWeightToleranceCheck
MaterialParameterTargetWeightToleranceUomCheck
MaterialParameterToleranceCheck
MaterialParameterToleranceUomCheck
MaterialParameterWeighingMaterialTypeForInlineWeighingCheck
MFCBomPositionsUsageCheck
MFCConfluenceOpenEndCheck
MFCFinalOutputsGenericCheck
MFCFinalOutputsMasterRecipeSpecificCheck
MFCMergeToFinalOutputNotInDispenseUPCheck
MFCPlannedQuantityEqualForCorrespondingTransfersCheck
MFCSwitchOpenEndCheck
MFCTransferNotIntoDispenseUPCheck
MFCUoMOfMaterialConvertionsDispenseCheck
NoLoopInProcedureGraphCheck
NoMFCOutputInInputOnlyMFCStructureCheck
NullStepsCheck
PhaseLibUsableInStructureTypeCheck
PrivilegeParametersCheck
ServerRunCapabilityPhaseCheck
ServerRunCapabilityStationCheck
ServerRunPhaseParameterCheck
SFCGraphCheck
TransitionConditionsExpressionsCheck
TransitionConditionsOfTransitionsBelowXorCheck
TriggerPhaseIsReferencedCheck
TriggerPhaseReferenceCheck
TriggerPhaseReferenceIdentifierCheck
TriggerPhaseReferenceUniqueCheck
WeighAndDispenseFlagCheck
WorkCenterStationAssignmentCheck
WorkflowAppendableDuringProcessingPossibleCheck
WorkflowCapabilityParameterCheck
WorkflowTypeCheck)

- This list can be extended if new check classes are added during system configuration and extension.
- New check classes must implement the *IRecipeStructureCheck* interface. In fact, it is recommended that new check classes are subclasses of the abstract *AbstractRecipeStructureCheck* base class.

- **Description**: A set of checks that is used to validate the structure and the properties of a master recipe, workflow, or building block generated by Recipe and Workflow Designer. The checks are performed when an object is validated. Mandatory checks are required to enable a correct execution within the Production Execution Client. We highly recommend not to disable these checks. By default, the following checks are available:

  - AutostartableCapabilityNotInSimultaneousBranchCheck
    Mandatory: no
    Checks that no **Auto-startable** capability is assigned to an operation located on a simultaneous branch.

  - BOMPositionNotEmptyCheck
    Mandatory: yes
    Checks that the MFC positions of direct MFC input items are not empty.

  - CapabilityCombinationCheck
    Mandatory: yes
    Checks for each capability parameter:

    - if it needs another capability in the same recipe element and
    - if it is compatible with all other capabilities in the same recipe element.

  - CapabilityEscalationDurationsCheck
    Mandatory: yes
    Checks for each capability parameter with the **Escalation-enabled** capability:

    - if at least one duration is set (due, overdue, expired),
    - when a duration is set, if the exception is enabled, and
    - if the entered durations are consistent (in ascending order: due < overdue < expired).

  - CapabilityRisksMatchedByPrivilegeUsagesCheck
    Mandatory: yes
    Checks that for each capability parameter with Exception enabled, a matching privilege parameter is defined (risk class must match the usage type of the privilege parameter).

  - CapabilityTriggerableHasChildCheck
    Mandatory: yes
    Checks that for a **Trigger-enabled** capability parameter at least one child (i.e. trigger phase reference) is defined.

- ChildrenCapabilititesCombinationCheck
  Mandatory: yes
  Checks that for each capability parameter of a recipe entity if it is compatible with all capabilities of the recipe entity's children.

- DispenseFlagCapabilityParameterCheck
  Mandatory: yes, enabled only for the PharmaSuite Dispense application
  Checks that none of the **Abort-and-reactivate-enabled**, **Auto-startable**, **Detachable**, **Escalation-enabled**, **Event-triggered**, **Server-run**, and **Trigger-enabled** capabilities is assigned to a Dispense operation.

- EquipmentPropertyRequirementIdentifierCheck
  Mandatory: yes
  Checks that the identifier of an equipment property requirement is not empty and unique within the equipment requirement parameter.

- EquipmentPropertyRequirementsCheck
  Mandatory: yes
  Checks that the property requirements of an equipment parameter match the equipment class/entity of that instance.

- EquipmentRequirementParameterCountsCheck
  Mandatory: yes
  Checks that the number of assigned equipment requirement parameters is valid.

- EquipmentRequirementParameterUniqueBOEPositionCheck
  Mandatory: yes
  Checks that each BOE position defined in the equipment requirement parameters is unique within the master recipe, master workflow, or building block.

- EquipmentRequirementUniqueClassCheck
  Mandatory: no
  Checks that each equipment class is unique within the equipment requirements of a phase.

- EquipmentRuleExpressionCheck
  Mandatory: yes
  Checks that the equipment requirement rule has an expression, there are no syntactic or semantic errors and no reference errors in the rule.

- ERPBomAllPositionsUsedInActualBomCheck
  Mandatory: no, enabled for Recipe Designer - Batch, Recipe Designer - Device
  If the master recipe is based on an ERP BOM, the system checks all MFC inputs of the master recipe against the ERP BOM items. Each material position in the ERP BOM must correspond to an MFC input with that position.

- ERPBomContainsAllActualBomPositionsCheck
  Mandatory: no, enabled for Recipe Designer - Batch, Recipe Designer - Device
  If the master recipe is based on an ERP BOM, the system checks all MFC inputs of the master recipe against the ERP BOM items. Each MFC input with a given position must correspond to the material in the ERP BOM.

- ERPBomItemConsistentMaterialParameterAttributesCheck
  Mandatory: no, enabled for Recipe Designer - Batch, Recipe Designer - Device
  If the master recipe is based on an ERP BOM, the system checks all of the materials assigned to any phase within the master recipe against the ERP BOM items. Each material with a given position must have the same common attributes (e.g. quantity, unit of measure) as the material in the ERP BOM.

- ERPBomPackagingLevelsCheck
  Mandatory: no, enabled for Recipe Designer - Batch
  If the master recipe is based on an ERP BOM, the system checks that the packaging levels of master recipe are equal to the packaging levels of the ERP BOM header.

- ERPBomQuantityCheck
  Mandatory: no, enabled for Recipe Designer - Batch
  If the master recipe is based on an ERP BOM, the system checks that the planned quantity of the master recipe is equal to the base quantity of the ERP BOM header (if there is an ERP BOM).

- ERPBomUniqueMFCPositionCheck
  Mandatory: no, enabled for Recipe Designer - Batch, Recipe Designer - Device
  The system checks that all MFC positions defined in the MFC-relevant material parameters of an execution path are unique.
  Duplicate positions that are also ERP BOM positions are reported differently.

- InputExpressionsCheck
  Mandatory: yes
  Checks that there are no invalid expressions that provide input values for parameter attributes within the underlying recipe structure.

- MasterRecipeMinMaxQuantityCheck
  Mandatory: yes, enabled for Recipe Designer - Batch, Recipe Designer - Device
  Checks the minimum/maximum quantity of the master recipe. It may be empty; but if it is not empty, a unit of measure must be assigned that can be converted to the unit of measure defined with the material's master data.

- MasterRecipePackagingInventoryLevelCheck
  Mandatory: yes
  Applies only to material output parameters that refer to the material (product) of the master recipe.
  Checks the inventory level attributes of the packaging levels of master recipes and their corresponding material output parameters.
  - A **Sublot** inventory level is defined only once.
  - A **Logistic unit** inventory level is defined only once.
  - A **Sublot and Logistic unit** inventory level is defined only once.
  - If a **Sublot and Logistic unit** inventory level is defined, no **Sublot** or **Logistic unit** inventory level is defined.
  - If both **Sublot** and **Logistic unit** inventory levels are defined, the **Sublot** inventory level is defined at a lower packaging level.

- MasterRecipePackagingLevelsConsistencyCheck
  Mandatory: yes
  Checks the consistency of the packaging levels of a master recipe (e.g. if the meaning is missing for a defined contained number).

- MasterRecipeQuantityCheck
  Mandatory: yes, enabled for Recipe Designer - Batch
  Checks the planned quantity of the master recipe. It must not be empty and a unit of measure must be assigned that can be converted to the unit of measure defined with the material's master data.

- MasterRecipeUsageTypeStructureCheck
  Mandatory: yes, enabled for Recipe Designer - Batch, Recipe Designer - Device
  If the usage type is **Cost center** (not **Production**): checks that the recipe

structure matches the following structure of one procedure, one unit procedure, and one operation.

- MaterialParameterActivePlannedPotencyForInlineCheck
  Mandatory: yes, applies to phase building blocks, see **InlineWeighingPhaseLibs** configuration key (page 112)
  For phases of an Inline Weighing operation, checks that a planned potency is defined for MFC-relevant material input parameters of the **Active** weighing material type.

- MaterialParameterAllowedWeighingMaterialTypeForInlineWeighingCheck
  Mandatory: yes, applies to phase building blocks, see **InlineWeighingPhaseLibs** configuration key (page 112)
  For phases of an Inline Weighing operation, checks the weighing material type.
  For ingoing MFC transfer items:
  - **Active** is **not** allowed,
  - **Compensator** and **Filler** are allowed if they were not set to one of these weighing material types before in at least one preceding material position.

- MaterialParameterAllowedWeighingMethodsForInlineWeighingCheck
  Mandatory: yes, applies to phase building blocks, see **InlineWeighingPhaseLibs** configuration key (page 112)
  For phases of an Inline Weighing operation, checks that **Pallet** weighing is not one of the allowed the weighing methods of MFC-relevant material parameters.

- MaterialParameterAsDefinedPlannedQuantityAndTolerancesForInlineWeighingCheck
  Mandatory: yes, applies to phase building blocks, see **InlineWeighingPhaseLibs** configuration key (page 112)
  For phases of an Inline Weighing operation, checks for each MFC-relevant material parameter with **As defined** as planned quantity mode that the planned quantity and tolerances are specified.

- MaterialParameterAsDefinedPlannedQuantityAndTolerancesForOutputWeighingCheck
  Mandatory: yes, applies to phase building blocks, see **OutputWeighingPhaseLibs** configuration key (page 113)
  For phases of an Output Weighing operation, checks for each MFC-relevant material parameter with **As defined** as planned quantity mode that the planned quantity, tolerances, and allowed weighing methods are specified.

- MaterialParameterAsProducedRequiresTolerancesCheck
  Mandatory: yes
  Checks for each MFC-relevant material parameter with **As produced** as planned quantity mode that tolerances are specified.

- MaterialParameterCountsCheck
  Mandatory: yes
  Checks that the number of assigned material parameters is valid.

- MaterialParameterDispenseCheck
  Mandatory: yes, enabled only for the PharmaSuite Dispense application
  Checks, for Dispense unit procedures, that the Dispense-specific attributes (allowed weighing methods, default weighing method, weighing type, planned potency, tolerances) are available and consistent for MFC-relevant material input parameters.

- MaterialParameterEmptyMFCPositionForInlineWeighingCheck
  Mandatory: yes, applies to phase building blocks, see **InlineWeighingPhaseLibs** configuration key (page 112)
  For phases of an Inline Weighing operation, checks that the MFC positions of MFC-relevant material input parameters are not empty. (The MFC positions of direct MFC inputs are covered by the **BOMPositionNotEmptyCheck** check (page 116).)

- MaterialParameterFixedQuantityCheck
  Mandatory: no
  - Checks for each MFC-relevant material parameter that the fixed quantity is not set for an output item.
  - Checks for each MFC-relevant material input parameter whose weighing material type is **Active** or **Compensator** that the fixed quantity is not set.

- MaterialParameterNumericMFCPositionCheck
  Mandatory: yes, enabled only for the PharmaSuite Dispense application
  Checks that the MFC positions of MFC-relevant input material parameters are not empty or are numeric, if at least one MFC-relevant material input parameter is a filler.

- MaterialParameterPackagingInventoryLevelCheck
  Mandatory: yes
  Applies only to material parameters that do not refer to the material (product) of the master recipe.
  Checks the inventory level attributes of the packaging levels of the material output parameters.

- A **Sublot** inventory level is defined only once.
- A **Logistic unit** inventory level is defined only once.
- A **Sublot and Logistic unit** inventory level is defined only once.
- If a **Sublot and Logistic unit** inventory level is defined, no **Sublot** or **Logistic unit** inventory level is defined.
- If both **Sublot** and **Logistic unit** inventory levels are defined, the **Sublot** inventory level is defined at a lower packaging level.

- MaterialParameterPackagingLevelsCheck
  Mandatory: no
  Checks that the packaging levels of the material parameter are equal to the packaging levels of the material (part).

- MaterialParameterPackagingLevelsConsistencyCheck
  Mandatory: yes
  Checks the consistency of the packaging levels of a material parameter (e.g. if the meaning is missing for a defined contained number).

- MaterialParameterPlannedPotencyUomCheck
  Mandatory: yes, enabled only for the PharmaSuite Dispense application
  Checks that the unit of measure of the planned potency of a material parameter is contained in the **UoMsOfPotency** list (see **UoMsOfPotency** configuration key (page 33)).

- MaterialParameterPlannedQuantityCheck
  Mandatory: yes
  Checks that each MFC-relevant material input parameter has a planned quantity, if its planned quantity mode is not **None**.

- MaterialParameterPlannedQuantityModeAsProducedCheck
  Mandatory: yes
  Checks for each MFC-relevant material parameter that the **As produced** planned quantity mode is set only for ingoing MFC transfer items. The **As produced** planned quantity mode is not allowed for MFC input, MFC output, and outgoing MFC transfer items.

- MaterialParameterPlannedQuantityModeForDispenseCheck
  Mandatory: yes
  Checks for each MFC-relevant material parameter that the planned quantity mode cannot be set to **None** for a Dispense unit procedure.

- MaterialParameterPlannedQuantityModeNoneCheck
  Mandatory: yes
  Checks for each MFC-relevant material parameter that the **None** planned quantity mode is set only for the **Auxiliary substance** weighing material type (or if no weighing material type is set). The **None** planned quantity mode is not allowed for the **Active**, **Compensator**, or **Filler** weighing material types.

- MaterialParameterPlannedQuantityModeOfTransferInForInlineWeighingCheck
  Mandatory: yes, applies to phase building blocks, see **InlineWeighingPhaseLibs** configuration key (page 112)
  For phases of an Inline Weighing operation, checks for each ingoing MFC transfer item that the **As produced** planned quantity mode is not set for the **Compensator** and **Filler** substance weighing material types.

- MaterialParameterQuantityUomCheck
  Mandatory: yes
  Checks that there is no invalid planned quantity assigned (i.e. without a UoM, with a UoM that cannot be converted to a target UoM, negative quantity).

- MaterialParametersWithSameMFCPositionConsistencyCheck
  Mandatory: yes
  Checks that for material parameters with the same MFC position all attributes are identical.

- MaterialParameterTargetWeightQuantityAndTolerancesCheck
  Mandatory: yes
  Only if a target weight has been defined for a material output parameter.
  Checks that the corresponding target weight tolerance (i.e. upper or lower) is also available.

- MaterialParameterTargetWeightQuantityUomCheck
  Mandatory: yes
  Only if a target weight has been defined for a material output parameter.
  Checks that there is no invalid target weight assigned (i.e. without a UoM, with a UoM that cannot be converted to the material's UoM, negative quantity).

- MaterialParameterTargetWeightToleranceCheck
  Mandatory: yes
  Only if a target weight has been defined for a material output parameter.
  If the material output parameter has one target weight tolerance (absolute or relative) defined (i.e. lower or upper): Checks that the corresponding target weight tolerance (i.e. upper or lower) is also available.

- MaterialParameterTargetWeightToleranceUomCheck
  Mandatory: yes
  Only if a target weight has been defined for a material output parameter.
  Checks that the target weight absolute tolerances are not negative and compatible with the unit of measure of the material and that the target weight relative tolerances are not negative and have % as unit of measure.

- MaterialParameterToleranceCheck
  Mandatory: yes
  If a material input parameter has one tolerance (absolute or relative) defined (i.e. lower or upper): Checks that the corresponding tolerance (i.e. upper or lower) is also available.

- MaterialParameterToleranceUomCheck
  Mandatory: yes
  Checks that absolute tolerances are not negative and compatible with the unit of measure of the material and the relative tolerances are not negative and have % as unit of measure.

- MaterialParameterWeighingMaterialTypeForInlineWeighingCheck
  Mandatory: yes, applies to phase building blocks, see **InlineWeighingPhaseLibs** configuration key (page 112)
  For phases of an Inline Weighing operation, checks that for each MFC-relevant material parameter a weighing material type is defined.

- MFCBomPositionsUsageCheck
  Mandatory: yes, applies only to procedure building blocks and master recipes
  Checks that the same BOM positions are used in each execution path.

- MFCConfluenceOpenEndCheck
  Mandatory: yes, applies only to procedure building blocks and master recipes
  Checks that for all MFC items each confluence has an incoming and an outgoing connector (or is merged with the final output).

- MFCFinalOutputsGenericCheck
  Mandatory: yes, applies only to procedure building blocks
  Checks that all unit procedures with MFC items are Dispense unit procedures.
  If not, then it checks that there are no unit procedures with non-final MFC output items and that, for each execution path, a final MFC output item is defined.

- MFCFinalOutputsMasterRecipeSpecificCheck
  Mandatory: yes, applies only to master recipes
  Checks that all unit procedures with MFC items are Dispense unit procedures.
  If not, then it checks that
  - there are no unit procedures with non-final MFC output items,
  - for each execution path, a final output item is defined,
  - the material of each final output item matches the material to be produced according to the master recipe, and
  - the quantity of each final MFC output matches the planned quantity to be produced by the master recipe.

- MFCMergeToFinalOutputNotInDispenseUPCheck
  Mandatory: yes, enabled only for the PharmaSuite Dispense application
  Checks that there is no confluence merged with the final output of a Dispense unit procedure.

- MFCPlannedQuantityEqualForCorrespondingTransfersCheck
  Mandatory: no
  Checks for each MFC-relevant material parameter that the planned quantities of the corresponding ingoing and outgoing MFC transfer items are equivalent.

- MFCSwitchOpenEndCheck
  Mandatory: yes, applies only to procedure building blocks and master recipes
  Checks that for all execution paths each switch has an incoming and an outgoing connector.

- MFCTransferNotIntoDispenseUPCheck
  Mandatory: yes, enabled only for the PharmaSuite Dispense application
  Checks that there are no MFC transfer items into a Dispense unit procedure.

- MFCUoMOfMaterialConvertionsDispenseCheck
  Mandatory: yes, enabled only for the PharmaSuite Dispense application
  Checks, for Dispense unit procedures, that the unit of measure of the output material must be convertible into the unit of measure of the referenced input material if the only allowed weighing method does not imply usage of a scale (e.g. Quantity entry).

- NoLoopInProcedureGraphCheck
  Mandatory: yes
  Checks if the graph does not contain loops between unit procedures in the procedure of a master recipe or building block.

- NoMFCOutputInInputOnlyMFCStructureCheck
  Mandatory: yes, enabled for Recipe Designer - Device, Workflow Designer
  Checks that no MFC outputs are defined for a device-specific master recipe or a master workflow.

- NullStepsCheck
  Mandatory: yes
  Checks that there are no dummy steps (null steps) in the recipe or workflow.

- PhaseLibUsableInStructureTypeCheck
  Mandatory: yes
  Checks that the usage context of a phase in a recipe structure matches the corresponding structure type.

- PrivilegeParametersCheck
  Mandatory: yes
  Checks that a usage type is assigned to each privilege parameter.
  Checks that the same usage type is not assigned to multiple privilege parameters.
  Checks that no privilege parameters with the **High** usage type are assigned to building blocks that have no pre-defined exceptions.
  Checks that neither the **Phase completion** nor the **Phase completion (sequential)** usage types are assigned to multiple privilege parameters.
  Checks for privilege parameters with the **Phase completion (sequential)** usage type that no double signature is configured.

- ServerRunCapabilityPhaseCheck
  Mandatory: yes
  Checks that all phases of an operation with the **Server-run** capability run on a server.
  Checks that none of the phases of an operation without the **Server-run** capability run on a server.

- ServerRunCapabilityStationCheck
  Mandatory: no
  Checks that no stations are assigned to operations with the **Server-run** capability.

- ServerRunPhaseParameterCheck
  Mandatory: no
  Checks that no privilege parameter with the **Phase completion**, **Phase completion (sequential)**, or *** (comment mandatory)** usage types is assigned to a phase of an operation with the **Server-run** capability.

- SFCGraphCheck
  Mandatory: yes
  Checks the structure of the entire SFC graph of the recipe for errors.

- TransitionConditionsExpressionsCheck
  Mandatory: yes
  Checks transition conditions that have been edited in Recipe and Workflow Designer so that - in a succeeding step - they can be transformed into transition conditions that can be evaluated by the activity set engine of FactoryTalk ProductionCentre.

- TransitionConditionsOfTransitionsBelowXorCheck
  Mandatory: yes
  Checks if each transition succeeding a selection branch has a non-empty transition condition.

- TriggerPhaseIsReferencedCheck
  Mandatory: no
  Checks that each ETO trigger phase is referenced by at least one child of a **Trigger-enabled** capability parameter (i.e. ETO template).

- TriggerPhaseReferenceCheck
  Mandatory: yes
  Checks that there are no invalid phase references in the children of a **Trigger-enabled** capability parameter (filled, (still) valid (available, suitable phase type), no external references, minimal path).

- TriggerPhaseReferenceIdentifierCheck
  Mandatory: yes
  Checks that the identifiers of trigger phase references are filled and unique within the **Trigger-enabled** capability parameter.

- TriggerPhaseReferenceUniqueCheck
  Mandatory: yes
  Checks that the phase references are unique for all children of a **Trigger-enabled** capability parameter.

- WeighAndDispenseFlagCheck
  Mandatory: yes, enabled only for the PharmaSuite Dispense application
  Checks that there is no unit procedure that contains a Dispense-flagged operation whose SFC (= Sequential Function Chart) graph has more than one operation step.

- WorkCenterStationAssignmentCheck
  Mandatory: yes
  Checks the consistency of the work center and station assignments to unit procedures and operations, respectively.

- WorkflowAppendableDuringProcessingPossibleCheck
  Mandatory: yes, enabled for Workflow Designer
  Checks that a master workflow can only be marked as **appendable during processing** if it is marked as **production-relevant**.

- WorkflowCapabilityParameterCheck
  Mandatory: yes, enabled for Workflow Designer
  Checks that the **Abort-and-reactivate-enabled** capability is not assigned to the unit procedure of a workflow.

- WorkflowTypeCheck
  Mandatory: yes, enabled for Workflow Designer
  Checks that a workflow type is defined.

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: At least all mandatory checks.

LibraryHolder/commons-versioning-impl.jar/versionTransitionSignature

- **Type**: Object - List
- **Value**: versionTransitionSignature
- **Description**: Defines a list of the signatures for status transitions.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/services-s88-ifc.jar/MaterialParameterCustomAttributesHandler

- **Type**: String
- **Value**:
  com.rockwell.mes.services.s88.ifc.recipe.MESMaterialParameterCustomAttributesHandler

- **Description**: Defines an implementation class for a customizable class to copy or initialize customer attributes related to material parameters in Recipe and Workflow Designer This applies to the "part (material) to material parameter" and "material parameter to ProcessBomItem" relations.
  The value must be set. It must be the fully qualified name of a java class implementing the *IMaterialParameterCustomAttributesHandler* interface. The class can extend the *MESMaterialParameterCustomAttributesHandler* standard class, and thus allows to change only one method.
  A method is called when a material parameter object is created in Recipe and Workflow Designer and a part (material) object is assigned. The second method is called when a ProcessBomItem object for the material parameter is created. The methods may copy attributes from the corresponding part (material) object to the material parameter object and from the material parameter object to the ProcessBomItem object.
  Additionally, attributes can be initialized. The method may only set custom attributes of the corresponding objects. Make sure to observe this restriction.
  The class must be added to the java class path. For this configuration, you need to create a JAR file containing the class and add it as a **Library** object in Process Designer. The changes will take effect when you restart the client.

  TIP
  In the current version of PharmaSuite, **ProcessBom** and **ProcessBomItem** objects are not visible, however they are still components of master recipes. The **ProcessBomItem** objects provide the data of the order step inputs. That is why the *copyCustomAttributes* handler is important to transfer data from material parameters (master data) to order step inputs (processing data).

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

services-s88-ifc.jar/FlexibleStateModelChangeRequestBatchRecipes

- **Type**: String
- **Value**: MESChangeRequestStatusGraph
- **Description**: Name of the flexible state model used for change requests of batch-specific master recipes.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for an **FSM** object in Process Designer.

services-s88-ifc.jar/FlexibleStateModelChangeRequestDiscreteRecipes

- **Type**: String
- **Value**: MESChangeRequestStatusGraph

- **Description**: Name of the flexible state model used for change requests of device-specific master recipes.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for an **FSM** object in Process Designer.

services-s88-ifc.jar/FlexibleStateModelChangeRequestMasterWorkflows

- **Type**: String
- **Value**: MESChangeRequestMasterWorkflowStatusGraph
- **Description**: Name of the flexible state model used for change requests of master workflows.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for an **FSM** object in Process Designer.

Configuration Keys for Data Manager

The following configuration keys are available for Data Manager.

Equipment/filterAvailableFSMsForPropertyType

- **Type**: Object - Filter
- **Value**: s88_eqm_availableFSMsForPropertyType
- **Description**: Defines the named filter for available FSMs of a property type.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Equipment/filterAvailableImagesForEquipmentClass

- **Type**: Object - Filter
- **Value**: s88_eqm_availableImagesForEquipmentClass
- **Description**: Defines the named filter for available images of an equipment class.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Equipment/filterAvailableSemanticPropertiesForFsmPropertyType

- **Type**: Object - Filter
- **Value**: s88_eqm_availableSemanticPropertiesForFSMPropertyType
- **Description**: Defines the named filter for available semantic properties of a property type.

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/ApplicationStart_DataManager/StartupChecks

- **Type**: List
- **Value**: N/A
- **Description**: List of configurable checks to be executed when Data Manager is started.
  Each check must implement the *IStartupCheck* interface.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for a **List** object in Process Designer.

Form/ApplicationStart_DataManager/StyleSheet

- **Type**: Object - List
- **Value**: mdt_StyleSheet
- **Description**: Defines the style sheet used in Data Manager.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/MasterDataEditor/BasicSearch.hideArchivedObjectsInitially

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether equipment objects in the **Archived** status are initially hidden in the search result of the **Basic search** mode of Data Manager - Equipment.
  If the value is set to **true**, Data Manager - Equipment - Basic Search hides equipment objects in the **Archived** status in its search results.
  If the value is set to **false**, Data Manager - Equipment - Basic Search displays equipment objects in the **Archived** status its search results. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/MasterDataEditor/BasicSearch.maximumNumberOfSelectedObjects

- **Type**: Long
- **Value**: 500

- **Description**: Specifies the maximum number of objects that can be selected at a time in Data Manager - Equipment - Basic Search.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Form/MasterDataEditor/mdt_buttonConfiguration.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the menu bar, toolbar, and context menus.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/MasterDataEditor/mdt_buttonConfiguration.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration of the menu bar, toolbar, and context menus is read from a file.
  If the value is set to **true**, Data Manager reads the XML configuration from the *mdtbuttonConfiguration.xml* file. This configuration is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
  If the value is set to **false**, Data Manager reads the XML configuration from the database (**mdt_buttonConfiguration** list).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/MasterDataEditor/mdt_defaultLayoutSettings.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the UI configuration of the floating windows.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/MasterDataEditor/mdt_defaultLayoutSettings.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the UI configuration of the floating windows is read from a file.
  If the value is set to **true**, Data Manager reads the configuration from the *mtd_defaultLayoutSetting.ini* file located in the user-specific directory. This configuration is recommended only for developers.
  If the value is set to **false**, Data Manager reads the configuration from the database (**mdt_defaultLayoutSetting** list).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/MasterDataEditor/ps_ExpressionEditorShortcuts.customerName

- **Type**: String
- **Value**: N/A
- **Description**: Defines the name of the XML configuration that is used to configure the shortcuts of the Expression editor.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/MasterDataEditor/ps_ExpressionEditorShortcuts.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the XML configuration of the shortcuts of the Expression editor is read from a file.
  If the value is set to **true**, Data Manager reads the XML configuration from the *ps_ExpressionEditorShortcuts.xml* file. This configuration is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
  If the value is set to **false**, Data Manager reads the XML configuration from the database (**ps_ExpressionEditorShortcuts** list).

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/MasterDataEditor/SmartSearch.includeArchivedGeneratedEntities

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether generated equipment entities in the **Archived** status are included in the search result of the **Smart search** mode of Data Manager - Equipment.
  If the value is set to **true**, Data Manager - Equipment - Smart Search includes generated equipment entities in the **Archived** status in its search results. This configuration may have an impact on the performance of Data Manager.
  If the value is set to **false**, Data Manager - Equipment - Smart Search does not include generated equipment entities in the **Archived** status its search results. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/MasterDataEditor/SmartSearch.includeNonArchivedGeneratedEntities

- **Type**: Boolean
- **Value**: True
- **Description**: Defines whether generated equipment entities in a status other than **Archived** are included in the search result of the Smart search mode of Data Manager - Equipment.
  If the value is set to **true**, Data Manager - Equipment - Smart Search includes generated equipment entities in a status other than **Archived** in its search result. This configuration may have an impact on the performance of Data Manager. This is the default setting.
  If the value is set to **false**, Data Manager - Equipment - Smart Search does not include generated equipment entities in a status other than **Archived** in its search result.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-s88equipment-ifc.jar/CapabilityValidator.Room

- **Type**: String
- **Value**: com.rockwell.mes.services.s88equipment.ifc.capabilityvalidation.IEquipmentCapabilityValidatorRoom

- **Description**: Defines the name of the validator interface used for the **Room (RS)** equipment type validation. Generally, the suffix after the dot (here: **Room**) corresponds to the meaning of the **EquipmentType** choice list element.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/services-s88equipment-ifc.jar/CapabilityValidator.Scales

- **Type**: String
- **Value**: com.rockwell.mes.services.s88equipment.ifc.capabilityvalidation.IEquipmentCapabilityValidatorScales
- **Description**: Defines the name of the validator interface used for the **Scale (RS)** equipment type validation. Generally the suffix after the dot (here: **Scales**) corresponds to the meaning of the **EquipmentType** choice list element.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Configuration Keys for the Exception Dashboard

The following configuration keys are available for the Exception Dashboard.

Form/ApplicationStart_ExceptionDashboard/StartupChecks

- **Type**: List
- **Value**: N/A
- **Description**: List of configurable checks to be executed when the Production Response Client is started.
  Each check must implement the *IStartupCheck* interface.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: Any valid name for a **List** object in Process Designer.

Form/ExceptionDashboard/prc_Additional_Filter_Attributes.customerName

- **Type**: String
- **Value**: prc_Additional_Filter_Attributes
- **Description**: Defines the name of the filter configuration that is used to configure additional filters in the Production Response Client.
  For this configuration, you need a development environment in order to modify

List objects in Process Designer or XML files located in the classpath (e.g. Eclipse).

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/ExceptionDashboard/prc_Additional_Filter_Attributes.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the filter configuration for additional filters is read from a file.
  If the value is set to **true**, the Production Response Client reads the filter configuration from the *prc_Additional_Filter_Attributes.xml* file. This configuration is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
  If the value is set to **false**, the Production Response Client reads the filter configuration from the database (**prc_Additional_Filter_Attributes** list).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Form/ExceptionDashboard/prc_Standard_Filter_Attributes.customerName

- **Type**: String
- **Value**: prc_Standard_Filter_Attributes
- **Description**: Defines the name of the filter configuration that is used to configure standard filters in the Production Response Client. They are always visible on the UI.
  For this configuration, you need a development environment in order to modify **List** objects in Process Designer or XML files located in the classpath (e.g. Eclipse).
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: A valid name for a **List** object in Process Designer.

Form/ExceptionDashboard/prc_Standard_Filter_Attributes.readFromFile

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the filter configuration for standard filters is read from a file. They are always visible on the UI.
  If the value is set to **true**, the Production Response Client reads the filter

configuration from the *prc_Standard_Filter_Attributes.xml* file. This configuration is recommended only for developers. For this configuration, you need a development environment in order to modify XML files located in the classpath (e.g. Eclipse).
If the value is set to **false**, the Production Response Client reads the filter configuration from the database (**prc_Standard_Filter_Attributes** list).

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/apps-exceptiondashboard-impl.jar/CustomizableColumnCreator

- **Type**: String
- **Value**: N/A
- **Description**: Defines an implementation class for a customizable column in the Overview panel of the Exception Dashboard. The column will be displayed as the first column. For more information, see chapter "Adding a Column to the Overview Panel of the Exception Dashboard" in Volume 3 of the "Technical Manual Configuration and Extension" [A4] (page 141).
  If a value is set, it must be the fully qualified name of a java class implementing the *ICustomizableColumnCreator* interface. The class must be added to the java class path. For this configuration, you need to create a JAR file containing the class and add it as a **Library** object in Process Designer. The changes will take effect when you restart the client.
  If no value is set, no additional column will be displayed. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/apps-exceptiondashboard-impl.jar/HideCapaID

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether CAPA ID is shown in the UI.
  If the value is set to **true**, the CAPA ID is shown in the UI.
  If the value is set to **false**, the CAPA ID is not visible in the UI. This is the default setting.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

Configuration Keys for Archive and Purge

The following configuration keys are available for archive and purge.

LibraryHolder/services-archivepurge-ifc.jar/AuditTrailTablesToPurge

- **Type**: Object - List
- **Value**: OBJECT_STATE_A
- **Description**: List of audit trail tables that are automatically purged if the associated table is purged.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/services-archivepurge-ifc.jar/DefaultArchiveDirectory-OrderBatch

- **Type**: String
- **Value**: <Windows default user directory>/Batch
- **Description**: Defines the default directory on your file system to which the batch record is exported for archiving. We recommend to configure the access rights to the directory to comply with your archiving procedures.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/services-archivepurge-ifc.jar/DefaultArchiveDirectory-OrderDevice

- **Type**: String
- **Value**: <Windows default user directory>/Device
- **Description**: Defines the default directory on your file system to which the device history record is exported for archiving. We recommend to configure the access rights to the directory to comply with your archiving procedures.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/services-archivepurge-ifc.jar/DefaultArchiveDirectory-OrderWorkflow

- **Type**: String
- **Value**: <Windows default user directory>/Workflow
- **Description**: Defines the default directory on your file system to which the workflow record is exported for archiving. We recommend to configure the access rights to the directory to comply with your archiving procedures.

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

LibraryHolder/services-archivepurge-ifc.jar/createCompressedRecordContainer

- **Type**: Boolean
- **Value**: False
- **Description**: Defines whether the record container (e.g. for batch records, device history records) is created as a compressed file.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: [False, True]

LibraryHolder/services-archivepurge-impl.jar/MaxPurgeChunkSize

- **Type**: Long
- **Value**: 100
- **Description**: Defines the maximum number of purge operations within a single transaction. The setting affects the size of the redo log file.
- **Evaluated**: Also when the **Change User** or **Register at Station** actions of the Production Execution Client are executed. The operator must not have been logged into the current Production Execution Client session before.
- **Range**: [1..1000]

service-archivepurge-impl.jar/PrerequisitesOrderArchive

- **Type**: Object - List
- **Value**: ap_PrerequisitesOrderArchive
- **Description**: List of prerequisites that must be fulfilled before an order or workflow can be exported for archive.
  By default, the following prerequisites are checked:
  - OrderArchivePrerequisiteOrderStatus
    Asserts that the order or workflow status allows the export for archive. The assertion is based on the **archivePurge.allow.Export** semantic property.
  - OrderArchivePrerequisiteParentOrdersOfAppendedWorkflowAreInTerminalState
    Asserts that all orders to which a workflow is appended have reached a final status. (This could be a logically final status, see **archivePurge.status.logicallyTerminal** semantic property.)

For a description of the semantic properties, please refer to chapter "Configuring Flexible State Models" in Volume 2 of the "Technical Manual Configuration and Extension" [A3] (page 141).

- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

service-archivepurge-impl.jar/PrerequisitesOrderPurge

- **Type**: Object - List
- **Value**: ap_PrerequisitesOrderPurge
- **Description**: List of prerequisites that must be fulfilled before an order or workflow can be purged.
  By default, the following prerequisites are checked:
  - OrderPurgePrerequisiteNoS88EquipmentBound
    Asserts that the order or workflow has no binding relation to S88 equipment any more.
  - OrderPurgePrerequisiteNoS88EquipmentWithGxPContext
    Asserts that the order has no current context relation to S88 equipment any more.
  - OrderPurgePrerequisiteOrderStatus
    Asserts that the order or workflow status allows to purge. This is the case if the status allows to export (see **service-archivepurge-impl.jar/PrerequisitesOrderArchive** configuration key (page 139)) or the status is marked with the **archivePurge.allow.PurgeWithoutExport** semantic property.
    For a description of the semantic properties, please refer to chapter "Configuring Flexible State Models" in Volume 2 of the "Technical Manual Configuration and Extension" [A3] (page 141).
  - OrderPurgePrerequisiteOrderExported
    Asserts that the order or workflow has been exported successfully unless the status allows purging without export or the workflow is not production-relevant.
  - OrderPurgePrerequisiteParentOrdersOfAppendedWorkflowAlreadyPurged
    Asserts that all orders to which a workflow is appended have been purged before the workflow is purged.
  - OrderPurgePrerequisiteNoIdentifiedSublots
    Asserts that there is no relation to a sublot that was identified for the order.
- **Evaluated**: Only when a PharmaSuite client is started.
- **Range**: N/A

Reference Documents

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

No.	Document Title	Part Number
A1	PharmaSuite Technical Manual Developing System Building Blocks	PSBB-PM007E-EN-E
A2	PharmaSuite Technical Manual Phases of the Dispense Package	PSDI-PM005E-EN-E
A3	PharmaSuite Technical Manual Configuration & Extension - Volume 2	PSCEV2-GR008E-EN-E
A4	PharmaSuite Technical Manual Configuration & Extension - Volume 3	PSCEV3-GR008E-EN-E
A5	PharmaSuite Technical Manual Installation - Enterprise Edition	PSEN-IN008E-EN-E
A6	PharmaSuite Technical Manual Administration	PSAD-RM008E-EN-E
A7	PharmaSuite Technical Manual Configuration & Extension - Volume 1	PSCEV1-GR008E-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 distributed with the FactoryTalk ProductionCentre installation.

No.	Document Title / Section
B1	Process Designer Online Help

TIP

To access the "Process Designer Online Help", use the following syntax: *http://<MES-PS-HOST>:8081/PlantOpsDownloads/docs/help/pd/index.htm*, where <MES-PS-HOST> is the name of your PharmaSuite server. To view the online help, the Apache Tomcat of the FactoryTalk ProductionCentre installation must be running.

The following documents are distributed with the PharmaSuite installation.

No.	Document Title / Section
C1	PharmaSuite-related Java Documentation: Interfaces of PharmaSuite

TIP

To access the "PharmaSuite-related Java Documentation", use the following syntax: *http://<MES-PS-HOST>:8080/PharmaSuite/javadoc/*, where <MES-PS-HOST> is the name of your PharmaSuite server.

The following third-party documentation is available online as reference:

No.	Document Title / Web Site
D1	Apache Logging Services Project: Apache log4j (http://logging.apache.org/log4j/)
D2	OSIsoft (http://www.osisoft.com/) PI JDBC 2010 R3 Administrator Guide Configuring PI Server Security
D3	ActiveMQ - Features - Message Features - ObjectMessage (http://activemq.apache.org/objectmessage.html)

Appendix A

Revision History

The following table describes the history of this document.

Changes related to the document:

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

Changes related to "Introduction" (page 1):

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

Changes related to "Extension and Naming Conventions" (page 7):

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

Changes related to "Managing Configurations" (page 11):

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

Changes related to "Logging and Debugging" (page 21):

Object	Description	Document
Configuring the Logging Capability (page 21)	Log file name of a Shop Operations Server updated to *<PharmaSuite event sheet name>-ftps.log*. Default location of log files for services updated to *<installDirectory>\installation\services\<PharmaSuite event sheet name>\bin\logs*.	1.0

Changes related to "Configuration Keys of PharmaSuite" (page 29):

Object	Description	Document
Configuration Keys of PharmaSuite (page 29)	Tip specific to the behavior of empty values added.	1.0
Configuration Keys Specific to the Production Management Client (page 40)	Added configuration keys: Form/ApplicationStart_ProductionManagementClient/StartupChecks, LibraryHolder/services-s88-impl.jar/SFCForceTransitionTimeout, LibraryHolder/services-s88-impl.jar/SFCTestForceTransitionTimeout	1.0
Configuration Keys Specific to the Production Execution Client (page 43)	Added configuration keys: Form/ApplicationStart_ProductionExecutionClient/ChangeUserOrStationChecks, Form/ApplicationStart_ProductionExecutionClient/StartupChecks, LibraryHolder/clientfw-commons.ifc.jar/StandardMouseTouchScrollingBehavior	1.0
Configuration Keys Specific to EBR (page 57)	Added configuration keys: LibraryHolder/apps-ebr-ifc.jar/CancelButtonEnabledOfExceptionDialog, LibraryHolder/apps-ebr-ifc.jar/PhaseRepairModeBlackList, LibraryHolder/services-wip-impl.jar/DCSDefaultProducedMaterialLocation	1.0
Configuration Keys for Recipe, Order, and Workflow Management (page 59)	Added configuration key: LibraryHolder/services-recipe-impl.jar/useGHSData	1.0
Configuration Keys for Barcodes (page 75)	Updated configuration keys: Forms/2DBarcodeImages/encodingByIText, LabelDesigns/2DBarcodeImages/encodingByIText	1.0
Configuration Keys for Exception Recording (page 90)	Added configuration keys: LibraryHolder/services-S88-impl.jar/ActivateExternalExceptionHandling, LibraryHolder/services-S88-impl.jar/CommunicationStatesToAllowManualExceptionSending, LibraryHolder/services-S88-impl.jar/MinimalRiskClassForExternalExceptionHandling, LibraryHolder/apps-ebr-ifc.jar/PhaseExecutionSkippedExcRisk, LibraryHolder/apps-ebr-ifc.jar/PhaseRepairModeExcRisk, LibraryHolder/apps-ebr-impl.jar/CanceledExceptionRisk, LibraryHolder/apps-ebr-impl.jar/UnloadOrderFromEBRExcRisk, LibraryHolder/services-eto-impl.jar/AutomaticExceptionOnResumeOfTemplateExcRisk,	1.0

Object	Description	Document
	LibraryHolder/services-eto-impl.jar/AutomaticExceptionOnFailureAtCreationOfRunOfETOExcRisk, LibraryHolder/services-order-ifc.jar/ForceTransitionExecutionOrderExcRisk, LibraryHolder/services-S88-ifc.jar/ExternalExceptionHandlingReceiver, LibraryHolder/services-S88-ifc.jar/ExternalExceptionHandlingSender	
Configuration Keys for Recipe and Workflow Designer (page 105)	LibraryHolder/apps-recipeeditor-impl.jar/RecipeStructureChecks configuration key: Added checks: BOMPositionNotEmptyCheck (replaces MaterialParameterEmptyMFCPositionCheck), ERPBomPackagingLevelsCheck, MasterRecipePackagingInventoryLevelCheck, MasterRecipePackagingLevelsConsistencyCheck, MaterialParameterPackagingInventoryLevelCheck, MaterialParameterPackagingLevelsCheck, MaterialParameterPackagingLevelsConsistencyCheck, MaterialParameterTargetWeightQuantityAndTolerancesCheck, MaterialParameterTargetWeightQuantityUomCheck, MaterialParameterTargetWeightToleranceCheck, MaterialParameterTargetWeightToleranceUomCheck, MaterialParametersWithSameMFCPositionConsistencyCheck, MFCBomPositionsUsageCheck, MFCConfluenceOpenEndCheck (replaces MFCItemOpenEndCheck), MFCFinalOutputsGenericCheck (replaces MFCOutputOnlyOneForProducedMaterialCheck, MFCFinalOutputCheck), MFCFinalOutputsMasterRecipeSpecificCheck (replaces MFCOutputOnlyOneForProducedMaterialCheck, MFCFinalOutputCheck), MFCSwitchOpenEndCheck, NoLoopInProcedureGraphCheck Removed check: MaterialParameterEmptyMFCPositionCheck (replaced by BOMPositionNotEmptyCheck), MFCFinalOutputCheck (replaced by MFCFinalOutputsGenericCheck, MFCFinalOutputsMasterRecipeSpecificCheck), MFCOutputOnlyOneForProducedMaterialCheck (replaced by MFCFinalOutputsGenericCheck, MFCFinalOutputsMasterRecipeSpecificCheck), MFCItemOpenEndCheck (replaced by MFCConfluenceOpenEndCheck), NoXorInProcedureGraphCheck Updated check: ERPBomUniqueMFCPositionCheck Added configuration keys:	1.0

Object	Description	Document
	AccessPrivilege/ConfidentialObject.Modifiable.WhenCreating RecipeStructure, Form/ApplicationStart_RecipeDesigner/StartupChecks Updated configuration key: LibraryHolder/services-s88-ifc.jar/MaterialParameterCustom AttributesHandler	
Configuration Keys for Data Manager (page 130)	Added configuration key: Form/ApplicationStart_DataManager/StartupChecks	1.0
Configuration Keys for the Exception Dashboard (page 135)	Added configuration key: Form/ApplicationStart_ExceptionDashboard/StartupChecks, LibraryHolder/apps-exceptiondashboard-impl.jar/HideCapaID	1.0

Index

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W