FT PharmaSuite®

INSTALLATION

RELEASE 10.02.00

TECHNICAL MANUAL

PUBLICATION PSES-IN010C-EN-E-SEPTEMBER-2021
Supersedes publication PSES-IN010B-EN-E

Rockwell Automation

Contact Rockwell See contact information provided in your maintenance contract.

Copyright Notice

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

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

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

Other Trademarks ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows 7 Professional, Windows 10, Windows Server 2008, Windows Server 2012, and Windows Server 2016 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

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

ControlNet is a registered trademark of ControlNet International.

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

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

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

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

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

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

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

Contents

Figures

Chapter 1

Introduction

This manual describes the installation of FT PharmaSuite® 10.02.00 on top of an installed FactoryTalk® ProductionCentre 10.4 MR7 (10.4.107937) environment with JBoss and how to set up the Live Data infrastructure and the Historian infrastructure. Please refer to the "PharmaSuite Supported Platforms Guide" [A4] (page 71) for exact version information.
For all information on installing your FactoryTalk ProductionCentre system, please refer to the FactoryTalk ProductionCentre installation documentation [A1], [A2], and [A3] (page 71). For the installation of PS Administration, please refer to "Implementation Guide PS Administration" [A10] (page 71).

The information is structured in the following sections:

- Installing PharmaSuite (page 3)
- Data Objects Provided Along with the Installation (page 37)
- Troubleshooting Installation Issues (page 41)
- Setting up the Live Data Infrastructure for PharmaSuite (page 43)
- Setting up the Historian Infrastructure for PharmaSuite (page 63)

Intended Audience

The manual is intended for administrators of a PharmaSuite system.

Due to the nature of the tasks that need to be performed during installation, the administrator should have the same experience level as required to install the FactoryTalk ProductionCentre platform.
The setup of a Live Data infrastructure requires a thorough knowledge in the area of automation integration.

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.

Installing PharmaSuite

IMPORTANT

PharmaSuite requires a pre-installed FactoryTalk ProductionCentre system. Please refer to the FactoryTalk ProductionCentre installation documentation [A3] (page 71) for installation instructions. It will guide you through all required steps to set up the FactoryTalk ProductionCentre system.

In the following, you will find a description of all system requirements that have to be met and all information you will need to collect prior to installing PharmaSuite on FactoryTalk ProductionCentre with JBoss.

Installation Prerequisites and Information

The following two checklists cover all preparatory steps and information required for installing PharmaSuite.

Prerequisites Checklist

Before you start the installation, check the prerequisites:

	Prerequisite	Your Notes	Done?
1	Adequate hardware and software as defined in "FactoryTalk ProductionCentre Release 10.4 Supported Platforms Guide" [A1] (page 71) and "PharmaSuite Supported Platforms Guide" [A4] (page 71).		
2	FactoryTalk ProductionCentre has been installed and the basic configuration according to the "FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced" [A3] (page 71) has been successfully performed.		
3	Only if you run multiple clients of different PharmaSuite versions on one computer: this setup requires a change of the default configuration of the download location during installation (see "Changing the Default Configuration" (page 5)).		

	Prerequisite	Your Notes	Done?
4	FactoryTalk ProductionCentre Support Home Page [B4] (page 72) has been checked for settings relevant to your system configuration (e.g. heap size for servers running on JBoss).		
5	Backups of the current FactoryTalk ProductionCentre databases have been created.		

6	Only if your PharmaSuite system operates with a Microsoft SQL database: Make sure that the **xp_cmdshell** proxy account is disabled.
	To verify the setting, run the following procedure in SQL Server Management Studio (SSMS):
	`USE master;`
	`GO`
	`EXEC sp_configure 'show advanced option'`
	The configuration value (config_value) must be 0. This means that Advanced Options are disabled.		

Information Checklist

Please prepare the information you will need during the installation process:

	Information	Your Notes	Done?
1	FactoryTalk ProductionCentre Application Server: IP address or DNS-resolvable name		
2	System configuration considers the following settings: The JBoss application server, SOS server, ActiveMQ, and the PharmaSuite upgrade engine run with the 64-bit version of Java 1.8.0_202. Each PharmaSuite client (including the AI server) is only supported for the 32-bit version of Java 1.8.0_202. Make sure that the specified programs use a JAVA_HOME environment variable pointing to the correct Java version.		
3	Only if you use the Microsoft On-screen Keyboard (OSK): For installation details, see section "Using an On-Screen Keyboard as Virtual Keyboard", chapter "Using Forms in the Production Execution Client" in Volume 1 of the "Technical Manual Configuration and Extension" [A11] (page 71).		

Changing the Default Configuration

PharmaSuite supports configurations where multiple clients of different PharmaSuite versions can run on one computer. However, this setup requires a change of the default configuration of the download location during installation.

For more information, please refer to chapter "Define the Download Location" in "FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced" [A3] (page 71) and chapter "Best Practices for Managing User Accounts of a Client Operating System" in "Technical Manual Administration" [A9] (page 71). Please also consider to set the *clientHome* property to the client's user profile directory (e.g. replace "C:" with a tilde character: *clientHome=~/.FTPC*). Thus,

- different users of the operating system can use the same workstation while keeping user-specific files separated,
- terminal services scenarios (e.g. Citrix) are supported, and
- issues caused by missing write access to *C:\.FTPC* are avoided.

If the default configuration was not changed during installation, it is not allowed to run multiple clients that belong to different PharmaSuite versions on one computer at the same time. This might cause corrupt system behavior and inconsistent log files.

Also, if the default configuration is not changed, but it is necessary to switch between clients of different PharmaSuite versions, the client-side file caches and log files have to be cleaned manually each time before another PharmaSuite version is started again.

Performing the Installation

To install PharmaSuite, perform the following steps:

1. Only if your PharmaSuite system operates with a Microsoft SQL database:
   Enable the READ_COMMITTED_SNAPSHOT option.
2. Enable object revisioning for all objects (page 7).
3. Adapt the JBoss configuration (page 7).
4. Download and expand the PharmaSuite Installer (page 8).
5. Optional:
   Configure audit trail settings (page 9).
6. Execute the installation script (page 11).
7. Create indexes on your database system, either for Microsoft SQL (page 13) or for Oracle (page 16).
8. Deploy the PharmaSuite help system (page 18).
9. Set up the ActiveMQ JMS message broker (page 19).

10. Optional:
    Configure the ActiveMQ JMS message broker failover scenario (page 22).
11. Set up the Shop Operations Server for PharmaSuite event sheets (page 23).
12. Install PharmaSuite phase building blocks (page 29).
13. Optional (if communication with Warehouse Management is required):
    Install and integrate EIHub (page 30).
14. Install Modular Framework-based applications (page 30).
    - PS Administration
    - Optional: Fit-for-purpose applications, such as Warehouse Management.
15. Remove inconsistencies from data dictionary classes (page 31).
16. Optional:
    Configure AI, EBR, OE, and TOM servers for the Shop Operations Server failover scenario (page 32).
17. Optional:
    Add support for heartbeat monitoring to non-standard Shop Operations Servers (page 33).

TIP

Further configuration options such as password complexity, the transfer and purge functionality, and the configuration of the Operational Data Store (ODS) are provided by FTPC Administrator. For more information, please refer to "FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced" [A3] (page 71) and "FactoryTalk ProductionCentre Administrator Release 10.4 User's Guide" [A5] (page 71).

Configuring Object Revisioning in FTPC Administrator

PharmaSuite requires **object revisioning** to be activated in FactoryTalk ProductionCentre. Object revisioning is disabled by default. To activate object revisioning, proceed as follows:

1. Refer to chapters "Configuring Database Logging" and "Object Revisioning Logging" in "FactoryTalk ProductionCentre Administrator Release 10.4 User's Guide" [A5] (page 71).
2. Open your installation-specific configuration as described in chapter "Connect to a Datasource and Configure Security" in "FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced" [A3] (page 71).
3. Set the **Object Revisioning Level** to **All Objects**.
4. Restart the JBoss Application Server to make the change take effect.
5. Continue with adapting the JBoss configuration (page 7).

Adapting the JBoss Configuration

To adapt the JBoss configuration, proceed as follows:

1. Navigate to the *standalone-full.xml* file in *<JBOSS_DIR>\standalone\configuration\*.
   - Identify the `<idle-timeout-minutes>` entry and set the default value to `15` (instead of `0`).
2. Adapt the JVM memory allocation pool parameter:
   Navigate to the *standalone.conf.bat* file in *<JBOSS_DIR>\bin\* and adapt the following line:
   ```
   JAVA_OPTS=-Xms1G -Xmx1G -XX:MetaspaceSize=96M
   -XX:MaxMetaspaceSize=256m
   ```
   to
   ```
   JAVA_OPTS=-Xms1G -Xmx2G
   ```
3. Extend JAVA_OPTS to add the *HeapDumpOnOutOfMemoryError* parameter:
   To the same line modified in step 2, add the following parameter
   `-XX:+HeapDumpOnOutOfMemoryError`
   This parameter enables the creation of a heap dump if an OutOfMemoryError occurs on the JBoss Application Server during execution. The heap dump helps with analyzing the root cause of the error. By default, a heap dump is created in a file called *java_pid<pid of the java process>.hprof* in the working directory of your JVM. You can specify an alternative file name or directory with the `-XX:HeapDumpPath=` option.

TIP
Please note that heap dump files are rather large, with the recommended settings for the JBoss Application Server they can reach a size of up to 2 GB. So, make sure there is enough hard disk space on the drive where the heap dump files will be generated. If you notice that an OOM error has occurred on the JBoss Application Server, restart the server and provide the heap dump file to the support team together with the log files.

4. Restart the JBoss Application Server to make the change take effect.
5. Continue with downloading and expanding the PharmaSuite Installer (page 8).

Downloading and Expanding the PharmaSuite Installer

The PharmaSuite Installer is available as an installation package on the Rockwell Automation Download Site.

To download and expand the installation package, proceed as follows:

1. Open Internet Explorer and navigate to the Rockwell Automation Download Site.
2. Navigate to the **PharmaSuite** section.
3. Select the PharmaSuite application to download.
4. On the Windows machine, expand the file that you have downloaded to extract the PharmaSuite Installer files to a target directory of your choice. From this directory, you will be able to execute the installation script.

TIP
Make sure the path to your target directory does not contain blanks.

5. Continue with the next step.
   - Only if you wish to modify the audit trail-related settings:
     Continue with configuring the audit trail settings (page 9).
   - Otherwise continue with executing the installation script (page 11).

Configuring Audit Trail Settings

PharmaSuite uses the generic audit trail mechanism of FactoryTalk ProductionCentre to keep track of changes to objects on the database level. If an object is modified, its previous representation is copied into an object-specific audit trail database table when the new object revision is written to the database. The Audit trail function in the Production Management Client makes use of audit trail database tables to display the corresponding data.

By default, PharmaSuite is configured to store audit trail data for all object types (see section "Configuring Object Revisioning in FTPC Administrator" (page 7)).
To avoid collecting unnecessary data and to reduce database growth, PharmaSuite allows to disable the collection of audit trail data on a per-object level.

The following rules apply when you disable the collection of audit trail data for specific objects:

- The installation script of PharmaSuite is pre-configured to disable the collection of audit trail data for specific objects.
- To change the settings, edit the *installer\Utils\conf\DisableAuditTrail.properties* file before you execute the installation script.
- Add a line with the name of a database table to the configuration file to disable the collection of audit trail data for this database table.
- Prefix comments with the hash symbol (#).

To disable audit trail for standard **FactoryTalk ProductionCentre objects**, proceed as follows:

- Add a line with the name of the object's database table to *DisableAuditTrail.properties*. If an object spans multiple database tables, all database tables related to this object have to be added to the exclusion list.
- Examples:
  Add the `MASTER_RECIPE` and `UDA_MasterRecipe` lines to disable the collection of audit trail data for **Master recipe** objects.
  Add the `INSTRUCTION` and `INSTRUCTION_BLOB` lines to disable the collection of audit trail data for **Work instruction** objects.

To disable audit trail for **AT objects**, proceed as follows:

- Add a line with the name of the object's database table to *DisableAuditTrail.properties*. The database tables of AT objects are prefixed with **AT_** followed by the name of the AT definition.
- Examples:
  Add the `AT_X_ObjectLock` line to disable the collection of audit trail data for **X_ObjectLock** AT objects.
  Add the
  ```
  AT_RS_PhDatGetTextValue0201
  AT_RS_PhOutGetTextValue0201
  AT_RS_ParamInstruction0200
  AT_RS_ParamExpValDef0200
  AT_RS_ParamExceptionDef0200
  AT_RS_ParamTextValid0201
  ```
  lines to disable the collection of audit trail data for the **phase data**, **phase output**, and **phase parameter** AT objects of the GetTextValue (RS) [2.1] phase building block.

When you have finished your changes, continue with executing the installation script (page 11).

Executing the Installation Script

To execute the downloaded and expanded installation script, proceed as follows:

1. Open Windows Explorer and navigate to the *cmds* subdirectory of the installer package (the directory to which you have expanded the PharmaSuite installer package).
2. Open the *settings.cmd* file for editing.
3. Adjust the following settings to match your system environment.
   - Replace `localhost` with an IP address or a DNS-resolvable name for your FactoryTalk ProductionCentre Application Server.
     You can keep the localhost default setting if your FactoryTalk ProductionCentre Application Server is on the same machine on which you plan to execute the installation scripts.
     - `JNPCONN`
       the JNP connection string of your FactoryTalk ProductionCentre Application Server. Example:
       `SET JNPCONN=remote+http://ftps01.myenterprise.com:8080`
     - `HTTPCONN`
       the HTTP connection string of your FactoryTalk ProductionCentre Application Server. Example:
       `SET HTTPCONN=http://ftps01.myenterprise.com:8080`
     - `JAVA_HOME`
       the Java environment setting of your FactoryTalk ProductionCentre client Application. Example:
       `SET JAVA_HOME=c:\32Bit\jdk1.8.0_202`
       Make sure that the `JAVA_HOME` path does not contain any blanks and that it is set according to section "Install the Required JDK", chapter "Third-Party Application Server Software" in "FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced" [A3] (page 71).

   Save and close the adjusted *settings.cmd* file.
4. Make sure the backups of the current FactoryTalk ProductionCentre databases have been created.

5. Open a command prompt.
   Start the installation script from the command prompt in the *cmds* directory:
   `install settings.cmd`
   - The script will request your confirmation for starting the installation process twice. To continue press any key. To abort the installation process press CTRL+C.
   - During the installation process, you can monitor the progress and the results of the various steps in the *logs* subdirectory of your installation package folder.
   - Depending on your machine's performance, the installation can take up to 15 minutes.
6. The installation script confirms the completion of the installation with **Finished successfully**.
7. Close the command prompt.
8. Restart the JBoss Application Server to make the configuration of the audit trail settings take effect.
9. Depending on the database system with which your PharmaSuite system operates, continue either with creating indexes on the Microsoft SQL Server database system (page 13) or with creating indexes on the Oracle database system (page 16).

Creating Indexes on the Microsoft SQL Server Database System

This section is only required if your PharmaSuite system operates with a Microsoft SQL database.
For improved performance of your system, a *Transact SQL* script file will create additional indexes. The file is provided along with the installer package.

To execute the *Transact SQL* script file on your FactoryTalk ProductionCentre production database, proceed as follows:

1. Open Windows Explorer and navigate to the *sql/mssql* subdirectory of the installer package.
2. Double-click the *CreateCustomIndexes.sql* script file to open it.

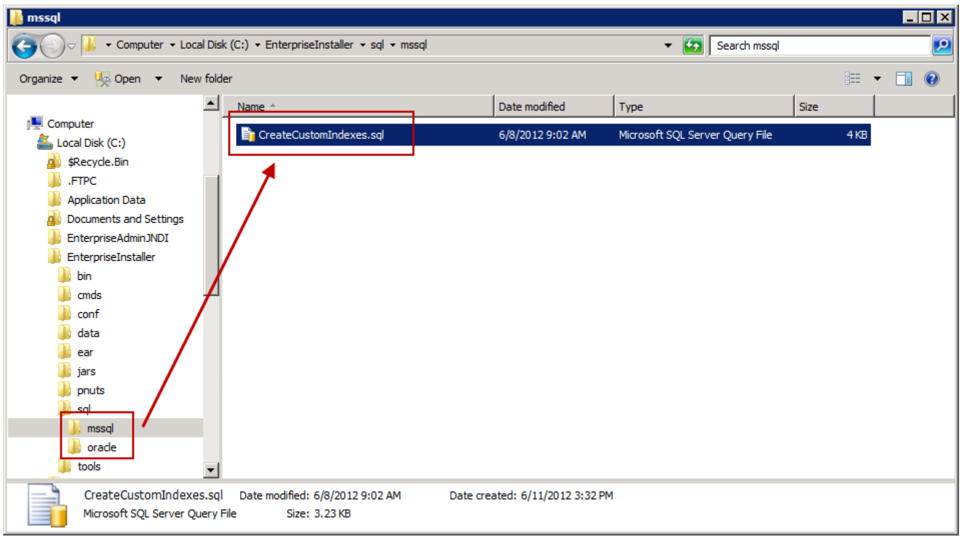
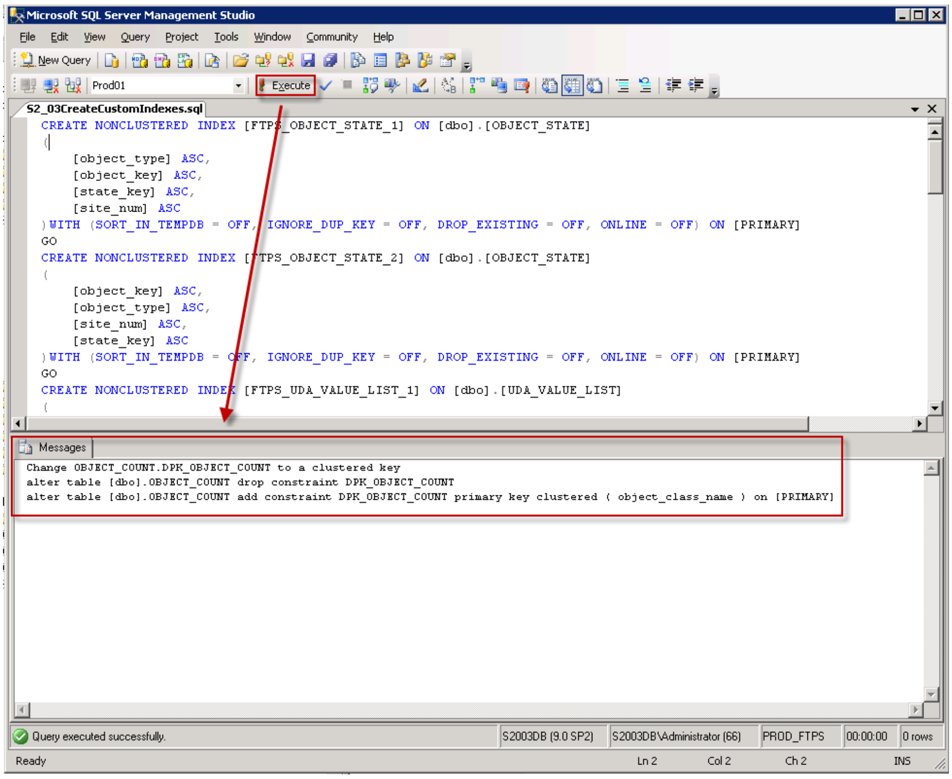


Figure 1: CreateCustomIndexes.sql script file for MS SQL

3. Now, Microsoft SQL Server Management Studio starts automatically and loads the script file.
   Connect to the server on which your production database is running.
4. Select your production database.
5. In the toolbar, click the **Execute** button to run the script.

6. After the successful execution of the script, the **Messages** pane lists the performed changes.


Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Community Help
New Query
Prod01
Execute
S2_03CreateCustomIndexes.sql
Messages
Query executed successfully.
S2003DB (9.0 SP2)
S2003DB\Administrator (66)
PROD_FTPS
00:00:00
0 rows
Ready
Ln 2
Col 2
Ch 2
INS

```
CREATE NONCLUSTERED INDEX [FTPS_OBJECT_STATE_1] ON [dbo].[OBJECT_STATE]
(
    [object_type] ASC,
    [object_key] ASC,
    [state_key] ASC,
    [site_num] ASC
)WITH (SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [FTPS_OBJECT_STATE_2] ON [dbo].[OBJECT_STATE]
(
    [object_key] ASC,
    [object_type] ASC,
    [site_num] ASC,
    [state_key] ASC
)WITH (SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF) ON [PRIMARY]
GO
CREATE NONCLUSTERED INDEX [FTPS_UDA_VALUE_LIST_1] ON [dbo].[UDA_VALUE_LIST]
(
```

```
Change OBJECT_COUNT.DPK_OBJECT_COUNT to a clustered key
alter table [dbo].OBJECT_COUNT drop constraint DPK_OBJECT_COUNT
alter table [dbo].OBJECT_COUNT add constraint DPK_OBJECT_COUNT primary key clustered ( object_class_name ) on [PRIMARY]
```

Figure 2: Execution of CreateCustomIndexes.sql for MS SQL

7. Execute the following SQL statements on your database to disable the trigger:

```
DISABLE TRIGGER [dbo].[dsOCIActivitySet] ON [dbo].[ACTIVITY_SET]
DISABLE TRIGGER [dbo].[dsOCDActivitySet] ON [dbo].[ACTIVITY_SET]
```

8. Close Microsoft SQL Server Management Studio.

9. Continue with enabling query store and activating forced query parameterization. (page 14)

Enabling Query Store and Activating Forced Query Parameterization on the Microsoft SQL Server Database System

This section is only required if your PharmaSuite system operates with a Microsoft SQL database.
In order to automatically capture a history of queries, plans, and runtime statistics for a review and force queries to be parameterized, proceed as follows:

1. Start Microsoft SQL Server Management Studio and connect to the server on which your production database is running.

2. Select your production database.

3. Create a new query using your current connection.

4. Execute the following SQL statements on your database to enable the query store:

```
ALTER DATABASE <database name> SET QUERY_STORE=ON
(OPERATION_MODE=READ_WRITE,MAX_STORAGE_SIZE_MB=1000,INTERVAL_LENGTH
_MINUTES=60,
CLEANUP_POLICY=(STALE_QUERY_THRESHOLD_DAYS
=30),SIZE_BASED_CLEANUP_MODE=AUTO,
QUERY_CAPTURE_MODE=AUTO,DATA_FLUSH_INTERVAL_SECONDS=900,MAX_PLANS_P
ER_QUERY=200);
```

TIP
This requires MS-SQL 2016 or higher.

5. Execute the following SQL statements on your database to activate a forced query parameterization:

```
ALTER DATABASE <database name> SET PARAMETERIZATION FORCED
```

TIP
This can be validated with a following query which returns “1” if “parametrization forced” is set:

```
SELECT name, is_parameterization_forced FROM sys.databases
```

6. Close Microsoft SQL Server Management Studio.

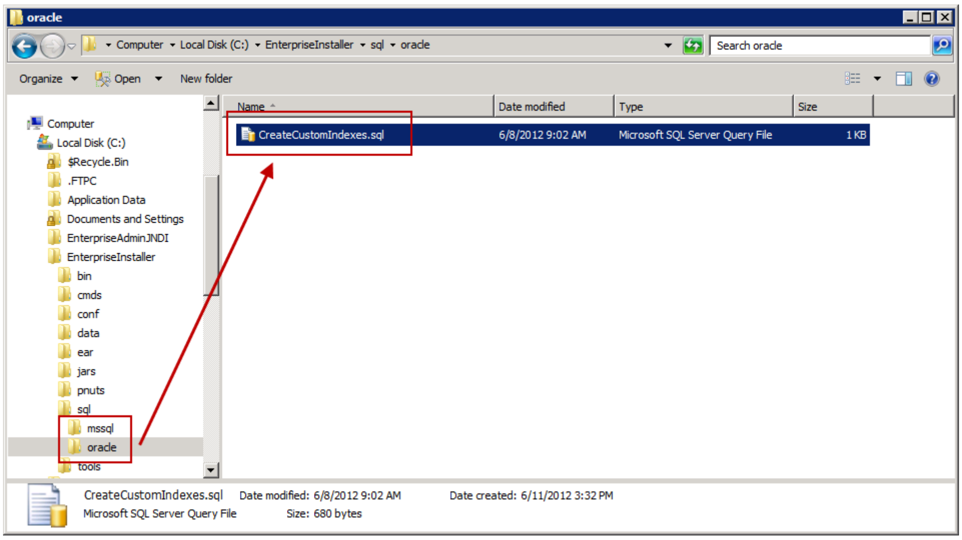
7. Continue with deploying the PharmaSuite help system. (page 18)

Creating Indexes on the Oracle Database System

This section is only required if your PharmaSuite system operates with an Oracle database.
For improved performance of your system, a *SQL* script file will create additional indexes. The file is provided along with the PharmaSuite Installer package.

To execute the *SQL* script file on your FactoryTalk ProductionCentre production database, proceed as follows:

1. Open Windows Explorer and navigate to the *sql/oracle* subdirectory of the PharmaSuite Installer package.
2. Open the *CreateCustomIndexes.sql* script file in a text editor.



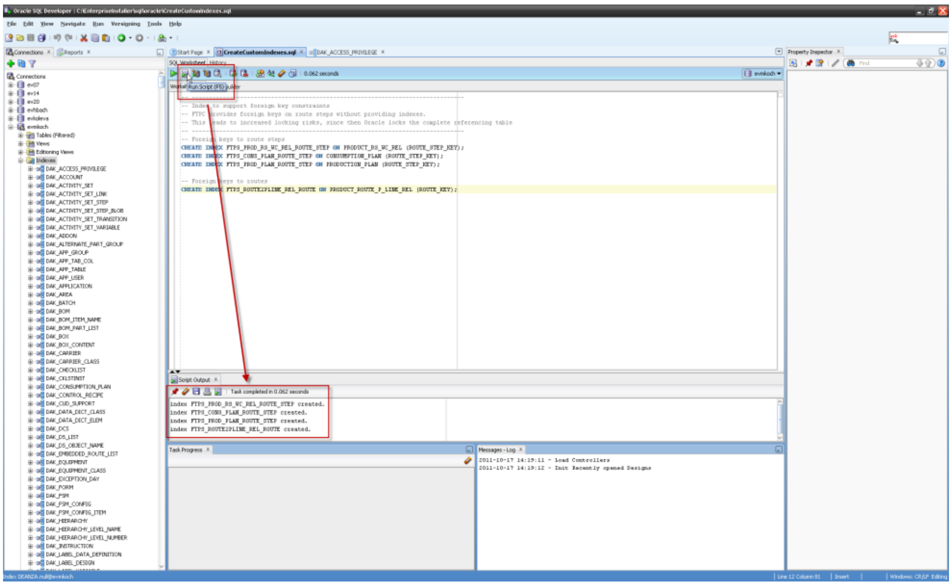
Name	Date modified	Type	Size
CreateCustomIndexes.sql	6/8/2012 9:02 AM	Microsoft SQL Server Query File	1 KB

CreateCustomIndexes.sql Date modified: 6/8/2012 9:02 AM Date created: 6/11/2012 3:32 PM
Microsoft SQL Server Query File Size: 680 bytes

Figure 3: CreateCustomIndexes.sql script file for Oracle

3. Start Oracle SQLDeveloper or another tool of your choice to connect to the database.
4. To connect to the database, use the same user you used to create the tablespace for the production database.
5. Copy the content of the *CreateCustomIndexes.sql* script file into a SQL Worksheet and execute the script.
6. In the toolbar, click the **Run Script (F5)** button to execute the script.

7. After the successful execution of the script, the **Script Output** tab displays **Task completed in ... seconds** and lists all indexes created.



```
-- ---------------------------------------------------------------------------
-- Index to support foreign key constraints
-- FTPC provides foreign keys on route steps without providing indexes.
-- This leads to increased locking risks, since then Oracle locks the complete referencing table
-- ---------------------------------------------------------------------------
-- Foreign keys to route steps
CREATE INDEX FTPS_PROD_RS_WC_REL_ROUTE_STEP ON PRODUCT_RS_WC_REL (ROUTE_STEP_KEY);
CREATE INDEX FTPS_CONS_PLAN_ROUTE_STEP ON CONSUMPTION_PLAN (ROUTE_STEP_KEY);
CREATE INDEX FTPS_PROD_PLAN_ROUTE_STEP ON PRODUCTION_PLAN (ROUTE_STEP_KEY);

-- Foreign keys to routes
CREATE INDEX FTPS_ROUTE2PLINE_REL_ROUTE ON PRODUCT_ROUTE_P_LINE_REL (ROUTE_KEY);
```

Task completed in 0.062 seconds

```
index FTPS_PROD_RS_WC_REL_ROUTE_STEP created.
index FTPS_CONS_PLAN_ROUTE_STEP created.
index FTPS_PROD_PLAN_ROUTE_STEP created.
index FTPS_ROUTE2PLINE_REL_ROUTE created.
```

```
2011-10-17 14:19:11 - Load Controllers
2011-10-17 14:19:12 - Init Recently opened Designs
```

Figure 4: Execution of CreateCustomIndexes.sql for Oracle

8. Close Oracle SQL Developer.

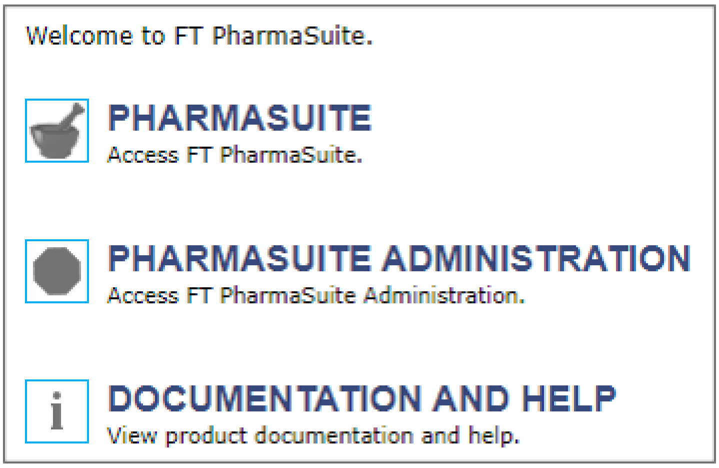
9. Continue with deploying the PharmaSuite help system (page 18).

Deploying the PharmaSuite Help System

The PharmaSuite start page and the online help system of PharmaSuite are included in an Enterprise Archive (EAR) file. This file needs to be deployed in the same manner as for the installation of the FactoryTalk ProductionCentre system. You can find the file in the *ear* subdirectory of the installer package.

To deploy the file, proceed as follows:

1. Copy the content of the *ear* subdirectory to the *deployments* subdirectory of your JBoss Application Server active configuration. This is the directory into which you have copied the *DSPlantOperations.ear* file.
2. As soon as the copy process is completed, the application server's EARDeployer immediately starts to work on the new file.
   Depending on your server's performance, the deployment process will take some time.
   You can monitor the progress at the JBoss console output or in the log files. Look for the line:
   ```
   [org.jboss.as.server.deployment] (MSC service thread ...:
   Starting deployment of "PharmaSuite-Help.ear" (runtime-name:
   "PharmaSuite-Help.ear")
   ```
   If the deployment is complete following lines appears in the log file:
   ```
   [org.jboss.web] (ServerService Thread Pool ... Register web
   context: /PharmaSuite [org.jboss.as.server]
   (DeploymentScanner-threads ... Deployed
   "PharmaSuite-Help.ear" (runtime-name :
   "PharmaSuite-Help.ear")
   ```
3. To verify the deployment, launch the PharmaSuite start page in your web browser (e.g. Internet Explorer). Use *PharmaSuite* instead of *PlantOperations* as address (e.g. *http://localhost:8080/PharmaSuite/*)



Welcome to FT PharmaSuite.

PHARMASUITE
Access FT PharmaSuite.

PHARMASUITE ADMINISTRATION
Access FT PharmaSuite Administration.

DOCUMENTATION AND HELP
View product documentation and help.

Figure 5: Verify deployment of PharmaSuite

4. Continue with setting up the ActiveMQ JMS message broker (page 19).

Setting up the ActiveMQ JMS Message Broker

For the JMS communication of the PharmaSuite clients with the Transition server, the EBR server, the Automation Integration server, the Triggered Operation Management server, and the Operation Execution server, PharmaSuite employs the widely used open source messaging server Apache ActiveMQ. The ActiveMQ Java process is typically run as a Windows service. It is the prerequisite for the usage of the **PharmaSuite_Transition_Server**, **PharmaSuite_EBR_Server**, **PharmaSuite_AI_Server**, **PharmaSuite_TOM_Server**, and **PharmaSuite_OE_Server** event sheets.

For more information, see ActiveMQ website [D1] (page 72).

To install the ActiveMQ JMS message broker, proceed as follows:

1. Download the windows binary distribution (i.e. apache-activemq-5.15.0.zip).
2. Extract the file (e.g. to C:\apache-activemq-5.15.0).
3. Adapt the ActiveMQ configuration for PharmaSuite:
   In *c:\apache-activemq-5.15.0\conf\activemq.xml*, perform the following steps:

   - Set up messaging port:
     In the **uri** attribute of the **<transportConnector ... />** tag, replace the original port of **61616** with the default PharmaSuite messaging port of **61646**. This should only be done for the **openwire (tcp)** protocol. Disable the entries of the other protocols.
     You may also choose another port, if you do so, you have to adapt the **MessageBrokerURL** configuration key in your **DefaultConfiguration** application (see chapter "Managing Configurations" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71)).

   - Adapt store limit and temporary store limit:
     In the **<storeUsage limit="..." />** tag, set the value to **10 gb**.
     In the **<tempUsage limit="..." />** tag, set the value to **5 gb**.

- Adapt the producer flow control and dead letter strategy to discard expired messages.
  These messages are no longer needed by PharmaSuite and would lead to out-of-memory situations if not discarded.
  In the existing **<policyEntries>** section, add this entry:

  ```
  <policyEntry queue=">" producerFlowControl="true">
  <!-- Tell the dead letter strategy not to process expired
       messages so that they will just be discarded instead
       of being sent to the DLQ. -->
  <deadLetterStrategy>
    <sharedDeadLetterStrategy processExpired="false" />
  </deadLetterStrategy>
  </policyEntry>
  ```

  Also add this entry in the existing <**policyEntry topic=**">" > section:

  ```
  <!-- Tell the dead letter strategy not to process expired
       messages so that they will just be discarded instead
       of being sent to the DLQ. -->
  <deadLetterStrategy>
    <sharedDeadLetterStrategy processExpired="false" />
  </deadLetterStrategy>
  ```

- Configure queues and topics to drop messages that have been sent to the dead letter queue. The messages in the dead letter queue are no longer needed by PharmaSuite and would lead to out-of-memory situations if not discarded.
  In the <**broker**> section add the following entry:

  ```
  <plugins>
    <discardingDLQBrokerPlugin dropAll="true"
      dropTemporaryTopics="true"
      dropTemporaryQueues="true"/>
  </plugins>
  ```

- Optional: Configure logging of discarded messages.
  For monitoring purposes, you may wish to activate the DEBUG level logging of the discardingDLQBrokerPlugin. This way you will be able to see which messages are being discarded. This may help you during issue investigation. However, keep in mind that this will generate extra logging, which will require additional space on your hard disk.
  To activate the logging, add the following section to the c:\apache-activemq-5.15.0\conf\log4j.properties file:

  ```
  # discarded messages
  log4j.appender.dlq=org.apache.log4j.RollingFileAppender
  log4j.appender.dlq.file=${activemq.data}/activemq_discard
  edToDLQ.log
  log4j.appender.dlq.maxFileSize=20MB
  log4j.appender.dlq.maxBackupIndex=20
  log4j.appender.dlq.append=true
  log4j.appender.dlq.layout=org.apache.log4j.PatternLayout
  log4j.appender.dlq.layout.ConversionPattern=%d | %-5p | %m
  | %c | %t%n
  log4j.logger.org.apache.activemq.plugin.DiscardingDLQBrok
  er=DEBUG,dlq
  log4j.additivity.org.apache.activemq.plugin.DiscardingDLQ
  Broker=false
  ```

- Optional: Disable message persistence
  Since PharmaSuite components do not rely on JMS message persistence, we recommend to disable it.
  However, if you are using failover or if any of your other components rely on message persistence, then you must not disable it.
  In order to disable the message persistence, in the **<broker.../>** tag, add the **persistent="false"** attribute.
  Below the **<persistenceAdapter>** tag, replace **<kahaDB directory="${activemq.data}/kahadb"/>** with **<memoryPersistenceAdapter/>**.

4. Adapt the service naming:
   In *c:\apache-activemq-5.15.0\bin\win64\wrapper.conf*, adapt the settings listed below as follows:

   ```
   wrapper.console.title=PharmaSuite ActiveMQ Broker
   wrapper.ntservice.name=PharmaSuite ActiveMQ Broker
   wrapper.ntservice.displayname=PharmaSuite ActiveMQ Broker
   wrapper.ntservice.description=PharmaSuite ActiveMQ Broker
   ```

5. Adapt the settings of the service wrapper related to log file size and number of rolled log files:
   In *c:\apache-activemq-5.15.0\bin\win64\wrapper.conf*, adapt the settings listed below as follows:
   ```
   wrapper.logfile.maxsize=1024k
   wrapper.logfile.maxfiles=5
   ```

6. In *c:\apache-activemq-5.15.0\conf\jetty.xml*, adapt the port setting:
   In the **<bean id="jettyPort" ..>** tag, change the value of the port from **value="8161"** to **value="8162"** (or another free port).

7. In *c:\apache-activemq-5.15.0\bin\win64\wrapper.conf*, make sure that `wrapper.java.command` references the correct Java version.
   For details, see item 2 in the "Information Checklist" (page 4).

8. Run *C:\apache-activemq-5.15.0\bin\win64\InstallService.bat* to install the ActiveMQ service.
   We recommend to start the task with the *Run as administrator* option enabled.

9. To start the ActiveMQ service, open a command prompt and type `sc start "PharmaSuite ActiveMQ Broker"`.

10. Continue with configuring the ActiveMQ JMS message broker failover scenario (page 22).

Configuring the ActiveMQ JMS Message Broker Failover Scenario

To configure the ActiveMQ JMS message broker failover scenario, proceed as follows:

1. Set up message brokers on two different machines.
   Example: **amqmasterserver** and **amqslaveserver**

2. Configure the message brokers for the master/slave usage. For details, see ActiveMQ - Features - Clustering - MasterSlave [D2] (page 72).

3. Set the value of the **MessageBrokerURL** configuration keys in your **DefaultConfiguration** application as follows:

   - **tcp://amqmasterserver:61646?connectionTimeout=2000,tcp://amqslaveserver:61646?connectionTimeout=2000**

   TIP
   PharmaSuite will automatically wrap this setting with the failover prefix using this pattern:
   ```
   failover:(<MessageBrokerURL>)?randomize=false&startupMaxReconnectAttempts=3&timeout=<MessagingTimeout>
   ```

4. Continue with setting up the Shop Operations Server (page 23) for PharmaSuite event sheets.

Setting up Shop Operations Servers

PharmaSuite uses the following event sheets: **PharmaSuite_Transition_Server**, **PharmaSuite_EBR_Server**, **PharmaSuite_AI_Server**, **PharmaSuite_TOM_Server**, and **PharmaSuite_OE_Server**.

For each event sheet, a Shop Operations Server must be set up according to the description given in chapter "Shop Operations Server" in "FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced" [A3] (page 71).

TIP

In order to prevent the automatic restart of a Shop Operations Server, update the *<SOS server>\vonf\wrapper.conf* file as follows:

1. In the **Wrapper General Properties** section, add the following lines
   ```
   #disable restart of JVM
   wrapper.disable_restarts.automatic=TRUE
   ```
2. In the Wrapper JVM Checks section, change the setting of the **wrapper.on_exit.-1** property from **RESTART** to **SHUTDOWN**:
   ```
   wrapper.on_exit.-1=SHUTDOWN
   ```

You can configure each Shop Operations Server individually (as described here) or configure one Shop Operations Server, copy this configuration, and adapt it for each further Shop Operations Server.

To assign the PharmaSuite event sheets in the Shop Operations Server, proceed as follows for each event sheet:

1. Set up the required number of Shop Operations Servers, one per event sheet.
   - Name the Shop Operations Servers similar to their event sheet, e.g. **PharmaSuite Transition server**, **PharmaSuite EBR server**, **PharmaSuite AI server**, **PharmaSuite TOM server**, and **PharmaSuite OE server**.
   - Users accessing the EBR, AI, Transition, TOM, and OE servers must be a member of the **MinimalAccess** and **PlantOpsOperator** user groups.
   - In the following, *<path_to_service>* is used as a placeholder for the location of your servers (e.g. *C:\Rockwell\PharmaSuite\installation\services\PharmaSuite_EBR_Server*).
2. Configure each Shop Operations Server with the Shop Operations Server administration console.
   - In the Shop Operations Server **Configuration – Configuring Server** page of each server, select and assign the corresponding event sheet.
   - In the Shop Operations Server **Configuration – Configure Logging** page of each server, disable the **Log Info Level** flag.

3. Configure the Java command (64-bit JDK and 32-bit for the AI server) for launching a JVM for each Shop Operations Server.

   - In each *<path_to_service>\conf\wrapper.conf* file, make sure that the command is set as follows:
     `wrapper.java.command=<Java installation path>/bin/java`
     Example:
     `wrapper.java.command=C:/64Bit/jdk1.8.0_202/bin/java`
   - You will continue to process the *<path_to_service>\conf\wrapper.conf* files in the next step.

4. Configure the maximum size of the heap space of each Shop Operations Server.

   - In each *<path_to_service>\conf\wrapper.conf* file, adapt the following line:
     `wrapper.java.additional.1=-Xmx512m`
     to
     `wrapper.java.additional.1=-Xmx2048m`
     Only for the AI server:
     `wrapper.java.additional.1=-Xmx1024m`
   - You will continue to process the *<path_to_service>\conf\wrapper.conf* files in the next step.

   TIP
   If you expect to have more than 1000 unit procedures running in parallel, increase the heap space to 4096 MB for the EBR and OE server.

5. Verify the settings related to thread pooling for each Shop Operations Server.

   - Make sure that in the **PharmaSuite_AI_Server**, **PharmaSuite_EBR_Server**, **PharmaSuite_OE_Server**, **PharmaSuite_TOM_Server**, and **PharmaSuite_Transition_Server** event sheets, the FactoryTalk ProductionCentre activity set, activity set step, and phase related to thread pooling is configured.
     This means that in each *<path_to_service>\conf\wrapper.conf*, the following line must be available (where **X** is the number of the configuration parameter):
     `wrapper.java.additional.X=-DActivitySetContainerWithThreadPool.numberOfThreads=50`
     For **X**, use the highest unused number of the `wrapper.java.additional` property. The configuration parameters are sequentially numbered (without gaps).
     Example:
     `wrapper.java.additional.11=-DActivitySetContainerWithThreadPool.numberOfThreads=50`
   - You will continue to process the *<path_to_service>\conf\wrapper.conf* files in the next step.

IMPORTANT

EBR server: A thread pool manages the execution of procedures and unit procedures of all orders processed on the EBR server. Hence, the number of orders processed at the same time is not restricted by the maximum of threads on the EBR server.

As the threads of the thread pool can be used by several procedures and unit procedures, a transition, e.g. between two operations, must not block the processing of the corresponding thread, since thread pooling may block the execution of further orders by the EBR server.

Blocking transitions can easily be introduced accidentally by using Pnuts code in the expression editor of Recipe and Workflow Designer when it includes blocking code, such as an infinite wait condition for some event. Therefore, we strongly recommend to refrain from using Pnuts code. Instead, use PharmaSuite functions only, since they return their result in a finite time.

However, blocking transitions can also occur due to the fact that transitions are only evaluated exactly once after each phase completion (within an operation run). This means that you must not create conditions that wait for a certain time or an external event to take place. What could happen in this situation is that at evaluation time the condition does not apply and thus blocks the transition, which then remains blocked as there is no further evaluation of the condition. If you wish to model a delay (directly or by waiting for an external event), we recommend to use a dedicated phase that periodically checks for the expected change in the condition.

OE server: The phase thread pool manages the execution of server-run phases in the OE server. Hence, the number of phases processed at the same time is not restricted by the maximum number of threads on the OE server. However it has to be ensured that those phases do not block their threads. This is considered as a programming error, the phase must use the schedule API call provided by PharmaSuite. For details, see chapter "Sever-side Client API for Phases" in "Technical Manual Developing System Building Blocks" [A8] (page 71).

TIP

Number of unit procedures: If you expect to have more than 1000 unit procedures running in parallel, increase the thread pool size to 100 for the EBR and OE server.

6. Configure the event sheet-specific wrapper log file for each Shop Operations Server.
   In each *<path_to_service>\conf\wrapper.conf* file:

   - To combine all logging information of an event sheet in a single log file, add the event sheet name and **-ftps** to the log file name configuration by adapting the following line (the file location is relative to the current directory):
     ```
     wrapper.logfile=../logs/wrapper.log
     ```
     Example:
     ```
     wrapper.logfile=../logs/PharmaSuite_EBR_Server-ftps.log
     ```

   TIP
   Please note that your changes cannot be applied if the configured file location does not exist.

   - Make sure the size of a wrapper log file is set to 10 Mbytes:
     ```
     wrapper.logfile.maxsize=10m
     ```
   - Make sure the Java service wrapper property is configured as follows:
     ```
     # Allow the service to interact with the desktop.
     wrapper.ntservice.interactive=false
     ```

7. Enable the creation of heap dump files on OutOfMemoryError:

   - To each *<path_to_service>\conf\wrapper.conf* file, add the following line:
     ```
     wrapper.java.additional.X=-XX:+HeapDumpOnOutOfMemoryError
     ```
     For **X**, use the highest unused number of the *wrapper.java.additional* property. The configuration parameters are numbered sequentially without gaps.
     This parameter enables the creation of a heap dump if an OutOfMemoryError occurs during execution. The heap dump helps with analyzing the root cause of the error. By default, a heap dump is created in a file called *java_pid<pid of the java process>.hprof* in the working directory of your JVM.
   - If you wish to change the folder where heap dump files are generated, add the following line:
     ```
     wrapper.java.additional.X=-XX:HeapDumpPath=<path to
     folder>
     ```
     We recommend to configure the heap dumps to be generated in the log folder. This way it will be easy to collect all resources needed for an issue analysis if an error occurs.

   TIP
   Please note that the heap dump files are rather large, depending on the configured heap size they can reach a size of up to 2 GB for the different Shop Operations Servers. So, make sure there is enough hard disk space on the drive where the heap dump files will be generated. If you notice that an OOM error has occurred on a Shop Operations Server, restart the service and provide the heap dump file to the support team together with the log files.

8. Optional: Configure logging specific to an event sheet for each Shop Operations Server.

    - Add a *log4j* configuration to *<path_to_service>\bin\logs\log4j_custom_local.properties*.
    - As a template use a suitable property file (e.g. *log4j_PharmaSuite_AI_Server.properties*, *log4j_PharmaSuite_EBR_Server.properties*, *log4j_PharmaSuite_TOM_Server.properties*, *log4j_PharmaSuite_OE_Server.properties*, *log4j_PharmaSuite_Transition_Server.properties*) located in the *installer\Utils\conf\* subdirectory of the PharmaSuite installer.

9. Optional (EBR server only): Configure the size of the thread pool used for the initial order loading.

    - The setting allows you to determine how the EBR server is going to initialize the running orders on startup. By default, the EBR server uses a thread pool of 50 threads to initialize the orders in parallel in order to optimize the start-up speed. To modify this setting, add the following line to the *<path_to_service>\conf\wrapper.conf* file of the EBR server:
      ```
      wrapper.java.additional.X=-DDistributedASInfoBundleForPro
      cedures.numberOfThreadsForCRRestart=<number of threads>
      ```
      For **X**, use the highest unused number of the `wrapper.java.additional` property. The configuration parameters are sequentially numbered (without gaps).
      Example:
      ```
      wrapper.java.additional.13=-DDistributedASInfoBundleForPr
      ocedures.numberOfThreadsForCRRestart=20
      ```
    - If you specify 1 or a smaller number, the thread pool will be disabled and the orders will be loaded sequentially.

10. Optional (EBR server only): Configure the size of the cache for optimization reasons.

    - By default, the cache size of the EBR server is set to 100 activity sets. If the cache is exhausted, the system has to fetch the activity set again, then at a lower speed.
      To modify this setting, add the following line to the *<path_to_service>\conf\wrapper.conf* file of the EBR server:
      ```
      wrapper.java.additional.X=-DEBRDistributedExecutionActivity.activitySetCacheSize=<number of activity sets>
      ```
      For **X**, use the highest unused number of the `wrapper.java.additional` property. The configuration parameters are sequentially numbered (without gaps).

    - To calculate the optimal size multiply the "maximum number of distinct active recipes" by "their average size (number of unit procedures)" and add some buffer.
      Example: 90 recipes * 42 unit procedures = 3780 activity sets
      ```
      wrapper.java.additional.14=-DEBRDistributedExecutionActivity.activitySetCacheSize=5000
      ```

    - A higher than necessary setting has no negative impact since the cache is cleaned up.

11. Optional (EBR server only): Enable load balancing by installing two or more nodes.

    - If you wish to improve the EBR scalability by enabling load balancing, you have to install two or more nodes as separate Shop Operations Server instances, based on the above given instructions.

    - If you install two or more nodes on the same computer, you have to create two separate installation directories, define different names for the Windows services, and specify different ports for Jetty and JMX.

    - For each node, add the following lines to its <path_to_service>\conf\wrapper.conf file:
      ```
      wrapper.java.additional.X-DLoadBalancingConfiguration.ServerNodeID=<node ID>
      wrapper.java.additional.X=-DLoadBalancingConfiguration.NumberOfNodes=<number of nodes>
      ```

IMPORTANT
The node ID must be a number between 1 and the total number of defined nodes.
The total number of nodes must be configured identically for each node.
So if you have three nodes, their IDs must be 1, 2, and 3 and the total number of nodes must be 3 in each configuration.

12. Only if an OE server is used.
    Define a station for the OE server.

    - The OE server must have a defined station. Per default, the **OE_Server** station is used. This station must be added to the generated XML configuration file of the event sheet. Proceed as follows:

    1. Adapt the XML tag in *<path_to_service>\bin\ShopOperationsServer.xml*:
        - Example:
          C:\Rockwell\PharmaSuite\installation\services\PharmaSuite_OE_Server \bin\ShopOperationsServer.xml
        - Adapt the XML tag for the **OE_Server** station:
          `<station>OE_Server</station>`
    2. Restart the event sheet service.

    TIP
    Alternatively, the station can be configured with the Shop Operations Server administration console. For details, see section "Configuring the Server", "Administering Shop Operations Server", chapter "Shop Operations Server" in "FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced" [A3] (page 71).

    The **OE_Server** station is provided with the system. You can configure your own station to be used by the OE server. This station must not have a work center assigned.
    If no station is defined for the OE server or a work center is assigned to the configured station, the service will stop its processing and provide an appropriate information in the log file.

13. Continue with installing PharmaSuite phase building blocks (page 29).

Installing PharmaSuite Phase Building Blocks

To install the PharmaSuite phase building blocks, please refer to "Technical Manual Installation - Building Blocks" of your phase package [A12] (page 71).

Continue with installing and integrating EIHub with PharmaSuite (page 30).

Installing and Integrating EIHub with PharmaSuite

To install EIHub and integrate it for communication with PharmaSuite, proceed as follows:

1. For general installation instructions, please refer to section "Running the Installer" of "Enterprise Integration Hub - For PharmaSuite Installation and Configuration Guide" [A15] (page 71).
2. To deploy the required WAR files, please refer to section "Installation" of "Enterprise Integration Hub - PharmaSuite/FactoryTalk Warehouse Management Integration Guide" [A16] (page 71).
3. To configure the communication between PharmaSuite and Warehouse Management, please refer to "Technical Manual Integration MES - WMS" [A17] (page 71).

Continue with installing Modular Framework-based applications (page 30).

Installing Modular Framework-based Applications

To install modular framework-based applications, proceed as follows:

1. For general installation instructions, please refer to "FactoryTalk ProductionCentre Deployment Manager Administration Guide" [A13] (page 71) and "FactoryTalk ProductionCentre Modular Framework WebSDK User's Guide" [A14] (page 71).
2. To install PS Administration, please refer to "Implementation Guide PS Administration" [A10] (page 71).
3. Optional:
   To install fit-for-purpose applications, such as Warehouse Management, please refer to the Implementation Guide of the respective fit-for-purpose application.

Continue with removing inconsistencies from data dictionary classes (page 31).

Removing Inconsistencies from Data Dictionary Classes

PharmaSuite is pre-configured to write log entries to document certain events of the system. Typical events are startup, shutdown of an application, malfunction of specific function.
One of the most frequently written log entries reports inconsistencies in the data dictionary, such as:

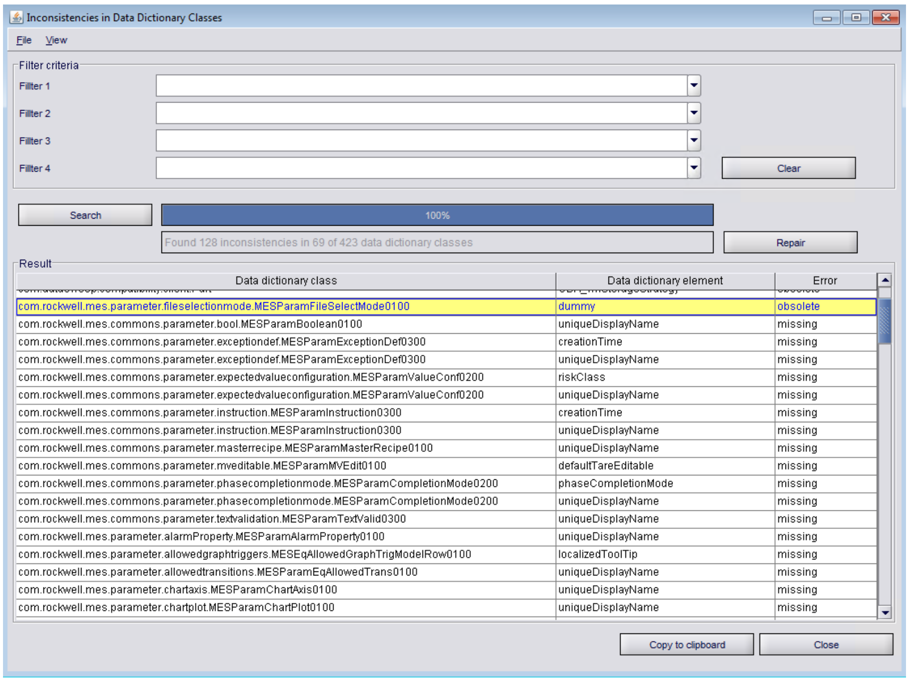
```
java.lang.Exception: Could not find the property for the data dictionary element
'uniqueDisplayName' in the class '
com.rockwell.mes.parameter.documenturl.MESParamDocumentUrl0100 '
    at com.rockwell.datadictionary.ExtendedBeanInfo.getDataDictPropertyDescriptors
       (ExtendedBeanInfo.java:424)
    at com.rockwell.datadictionary.ExtendedBeanInfo.<init>(ExtendedBeanInfo.java:118)
```

In order to avoid log entries of this kind and potential performance degradation, you can remove the inconsistencies from the corresponding data dictionary classes with a specific tool provided along with PharmaSuite.

To remove inconsistencies from data dictionary classes, proceed as follows:

1. In Process Designer, run the **mes_DataDictConsistencyChecker** form to start the consistency checker.
2. Click the **Search** button. In the **Results** panel, the form displays the data dictionary classes with inconsistencies.

TIP
Use the filters to narrow down the list of data dictionary classes displayed and processed by the **Repair** button.



Found 128 inconsistencies in 69 of 423 data dictionary classes

Data dictionary class	Data dictionary element	Error
com.rockwell.mes.parameter.fileselectionmode.MESParamFileSelectMode0100	dummy	obsolete
com.rockwell.mes.commons.parameter.bool.MESParamBoolean0100	uniqueDisplayName	missing
com.rockwell.mes.commons.parameter.exceptiondef.MESParamExceptionDef0300	creationTime	missing
com.rockwell.mes.commons.parameter.exceptiondef.MESParamExceptionDef0300	uniqueDisplayName	missing
com.rockwell.mes.commons.parameter.expectedvalueconfiguration.MESParamValueConf0200	riskClass	missing
com.rockwell.mes.commons.parameter.expectedvalueconfiguration.MESParamValueConf0200	uniqueDisplayName	missing
com.rockwell.mes.commons.parameter.instruction.MESParamInstruction0300	creationTime	missing
com.rockwell.mes.commons.parameter.instruction.MESParamInstruction0300	uniqueDisplayName	missing
com.rockwell.mes.commons.parameter.masterrecipe.MESParamMasterRecipe0100	uniqueDisplayName	missing
com.rockwell.mes.commons.parameter.mveditable.MESParamMVEdit0100	defaultTareEditable	missing
com.rockwell.mes.commons.parameter.phasecompletionmode.MESParamCompletionMode0200	phaseCompletionMode	missing
com.rockwell.mes.commons.parameter.phasecompletionmode.MESParamCompletionMode0200	uniqueDisplayName	missing
com.rockwell.mes.commons.parameter.textvalidation.MESParamTextValid0300	uniqueDisplayName	missing
com.rockwell.mes.parameter.alarmProperty.MESParamAlarmProperty0100	uniqueDisplayName	missing
com.rockwell.mes.parameter.allowedgraphtriggers.MESEqAllowedGraphTrigModelRow0100	localizedToolTip	missing
com.rockwell.mes.parameter.allowedtransitions.MESParamEqAllowedTrans0100	uniqueDisplayName	missing
com.rockwell.mes.parameter.chartaxis.MESParamChartAxis0100	uniqueDisplayName	missing
com.rockwell.mes.parameter.chartplot.MESParamChartPlot0100	uniqueDisplayName	missing

Figure 6: Inconsistencies in Data Dictionary Classes

3. The entries in the **Error** column have the following meaning:
   - **obsolete**: Data dictionary elements can be removed from the data dictionary class.
   - **missing**: Data dictionary elements must be added to the data dictionary class.

> TIP
> If you wish to retain the list of data dictionary classes, click the **Copy to clipboard** button to copy the list and then paste it into a system-external text editor. We recommend to archive the list.

4. Click the **Repair** button to automatically remove the inconsistencies from the data dictionary classes listed in the **Result** panel.
   For all automatically created data dictionary elements, the **Hidden** property is set to **Yes**.
5. Close the form.
6. Close Process Designer.
7. Continue with the next step.
   - Only if you wish to modify the Shop Operations Server Failover Scenario: Continue with configuring AI, EBR, OE, and TOM servers for Shop Operations Server failover scenario (page 32).
   - Only if you wish to perform heartbeat checks for non-standard Shop Operations Servers:
     Continue with adding support for heartbeat monitoring to non-standard Shop Operations Servers (page 33).
   - Otherwise continue with verifying the installation (page 34).

Configuring AI, EBR, OE, and TOM Servers for Shop Operations Server Failover Scenario

The EBR server and the TOM server support failover by using the FactoryTalk ProductionCentre Shop Operations Server Master/Slave failover mechanism.

To configure the failover, proceed as follows:

1. Set up two Shop Operations Server instances on two different machines.
2. Follow the instructions in section "Configuring Failover" in chapter "Shop Operations Server" in "FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced" [A3] (page 71).

3. If you have installed multiple EBR nodes, you need to set up a separate slave instance for each node. For details on setting up multiple EBR nodes, see section "Setting up Shop Operations Servers" (page 23), step 10.
4. Continue with the next step.
   - Only if you wish to perform heartbeat checks for non-standard Shop Operations Servers:
     Continue with adding support for heartbeat monitoring to non-standard Shop Operations Servers (page 33).
   - Otherwise continue with verifying the installation (page 34).

Adding Support for Heartbeat Monitoring to Non-standard Shop Operations Servers

All PharmaSuite Shop Operations Servers support the server heartbeat monitoring mechanism (see section "Basic Operations" in "Production Execution User Documentation" [C1] (page 72).

However, other Shop Operations Servers in your PharmaSuite environment (e.g. for ERP integration) can be included in this monitoring. For this purpose, you have to adapt your configuration.

To add heartbeat support, proceed as follows:

1. In your event sheet, add support for messaging if it is not already done:
   In the **Event Explorer**, click **Event Actions**, select **Add Activity**, and then **MessagingActivity**.
2. In your event sheet, add startup and shutdown callbacks:
   In the **EventSheetDefinitions**, select the **afterStart** function and add the following lines:

```
import("com.rockwell.mes.commons.messaging.ifc.serverheartbeat
       .ServerHeartbeatProducer")
ServerHeartbeatProducer::INSTANCE.startup()
```

   In the **EventSheetDefinitions**, select the **beforeStop** function and add the following two lines:

```
import("com.rockwell.mes.commons.messaging.ifc.serverheartbeat
       .ServerHeartbeatProducer")
ServerHeartbeatProducer::INSTANCE.shutdown()
```

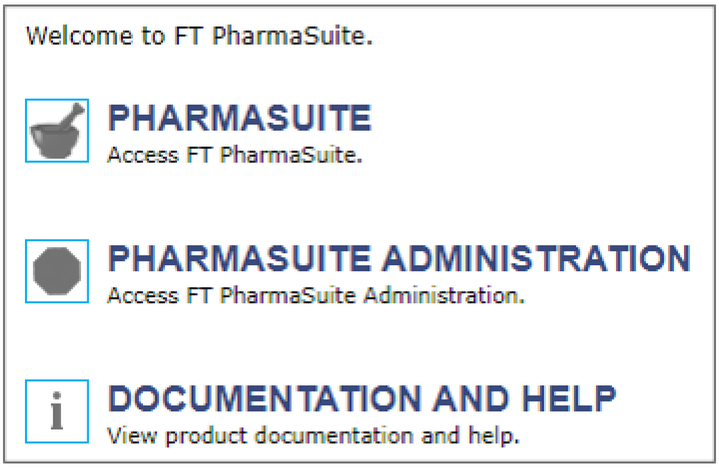
3. Add your event sheet to the configured list of mandatory servers:
   In the **MandatoryServerEventSheets** list, add the name of your event sheet (see **MandatoryServerEventSheets** configuration key in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71).
4. Continue with verifying the installation (page 34).

Post-installation Steps

After you have successfully installed and configured PharmaSuite on your system, you need to verify the installation.

Verifying the Installation

To verify the PharmaSuite installation, open the start page (http://<MES-PS-HOST>:8080/PharmaSuite/) in a web browser (e.g. Internet Explorer), and click the PHARMASUITE link in the browser window.
<MES-PS-HOST> is the name of your PharmaSuite server.



Welcome to FT PharmaSuite.

PHARMASUITE
Access FT PharmaSuite.

PHARMASUITE ADMINISTRATION
Access FT PharmaSuite Administration.

DOCUMENTATION AND HELP
View product documentation and help.

Figure 7: Verify deployment of PharmaSuite

Now you can log in either as PMC administrator or as PEC administrator. A PEC administrator also has the necessary rights to the Exception Dashboard in the Production Responses application.
For the PMC administrator role, use the pmcadmin login and pmcadmin password.
For the PEC administrator role, use the pecadmin login and pecadmin password.

Accessing PharmaSuite Servers

The default user that gets assigned during installation is the **shopopsserver** user. For the password, please contact the Rockwell Automation Support.
Users accessing the EBR, AI, Transition, TOM, and OE servers must be a member of the **MinimalAccess** and **PlantOpsOperator** user groups.
For further information on user rights for FactoryTalk Live Data, which is used in automation integration, please refer to chapter "Setting up the Live Data Infrastructure for PharmaSuite" (page 43).

Ensuring Synchronized Date and Time Settings

Make sure that the date and time settings are synchronized between the machines that run the PharmaSuite clients, PharmaSuite servers, and the ActiveMQ JMS message broker.

We recommend to

- use the Active Directory mechanism for time synchronization (if available) or
- set up an automated task to frequently synchronize the times. For example, run `hourly:schtasks /create /TN HourlyTimeSync /SC HOURLY /RU SYSTEM /TR "W32tm.exe /resync"` as administrator.
  The **Windows Time** service must be started. Its **Startup Type** should be set to **Automatic**.
  All systems should be synchronized with the same NTP server.

Security Considerations for Windows Services

To prevent privileges escalations, make sure that Windows services are configured to run on a Local Service account. For the AI server, keep in mind that its Windows account requires access to Factory Live Data or the Historian system.

To prevent disclosing sensitive information through read access for Windows users who belong to the Users group, make sure that the executable files, directory objects, and other associated product artifacts (i.e. configuration files) for Windows services must be installed with access control lists that only allow administrators or Local Service accounts to read and modify the folders, their contained sub-folders, and files. This will also prevent the placing of malicious files that might be loaded by the service.

To adapt the access control lists of the PharmaSuite Windows services on the services' installation folders, a PowerShell script `setFolderPermissions.ps1` is provided. For using it, proceed as follows:

1. Make sure that none of the services are running.
2. Open Windows PowerShell as Administrator.
3. Execute the following commands on the services' folders:
   - `Import-Module .\setFolderPermissions.ps1`
     If the script is located in the PowerShell directory
     or
     `Import-Module <full qualified path to the script> setFolderPermissions.ps1`
   - `Set-FolderPermissions -Path <service-folder-name>`

Chapter 3

Data Objects Provided Along with the Installation

Along with the software, the following data objects are installed, which are available in Data Manager

	Object Type	Identifier	Details
1	Equipment class	Container_RS_1	Default class for containers.
2	Equipment class	Room_RS_1	Default class for rooms.
3	Equipment class	Scale_RS_1	Default class for scales.
4	Equipment graph	ContainerCleaning_RS_1	Default graph for container cleaning, included with import of the **Container_RS_1** class.
5	Equipment graph	RoomCleaning_RS_1	Default graph for room cleaning, included with import of the **Room_RS_1** class.
6	Equipment graph	ScaleCalibration_RS_1	Default graph for scale calibration, included with import of the **Scale_RS_1** class.
7	Equipment graph	ScaleTest_RS_1	Default graph for scale test, included with import of the **Scale_RS_1** class.
8	Property type	Container Clean Shelflife (RS)	Specification property type of the Duration data type, included with import of the **Container_RS_1** class or the **ContainerCleaning_RS_1** graph, respectively.
9	Property type	Container Sublot (RS)	Runtime property type of the String data type and the **Current Sublot (RS)** purpose, included with import of the **Container_RS_1** class or the **ContainerCleaning_RS_1** graph, respectively.

	Object Type	Identifier	Details
10	Property type	Container Tare (RS)	Runtime property type of the MeasuredValue data type and the **Current Tare (RS)** purpose, included with import of the **Container_RS_1** class.
11	Property type	Container Type (RS)	Specification property type of the **Container (RS)** equipment type, included with import of the **Container_RS_1** class.
12	Property type	General Clean Shelflife (RS)	Specification property type of the Duration data type, included with import of the **Room_RS_1** class or the **RoomCleaning_RS_1** graph, respectively.
13	Property type	Room Cleaning Rules (RS)	Specification property type of the **RoomCleaningRules** data type, included with import of the **Room_RS_1** class.
14	Property type	Room Type (RS)	Specification property type of the **Room (RS)** equipment type, included with import of the **Room_RS_1** class.
15	Property type	Scale Calibration Shelflife (RS)	Specification property type of the Duration data type, included with import of the **Scale_RS_1** class or the **ScaleCalibration_RS_1** graph, respectively.
16	Property type	Scale Configuration (RS)	Specification property type of the ScaleConfiguration data type, included with import of the **Scale_RS_1** class.
17	Property type	Scale Load (RS)	Runtime property type of the String data type and the **Current load (RS)** purpose, included with import of the **Scale_RS_1** class.
18	Property type	Scale Ranges (RS)	Specification property type of the Ranges data type and the **ScaleRanges (RS)** purpose, included with import of the **Scale_RS_1** class.

	Object Type	Identifier	Details
19	Property type	Scale Test and Calibration (RS)	Specification property type of the ScaleTestAndCalibration data type, included with import of the **Scale_RS_1** class.
20	Property type	Scale Test Shelflife (RS)	Specification property type of the Duration data type, included with import of the **Scale_RS_1** class or the **ScaleTest_RS_1** graph, respectively.
21	Property type	Scale Type (RS)	Specification property type of the **Scale (RS)** equipment type included with import of the **Scale_RS_1** class.
22	Property type	Work Center Assignment (RS)	Specification property type of the WorkCenterAssignment data type, included with import of the **Scale_RS_1** class or **Room_RS_1** class, respectively.

Chapter 4

Troubleshooting Installation Issues

If issues occur during the installation process that prevent the installation from completing successfully, you will usually be notified by the system. It is recommended, however, that you verify the successful installation by viewing the log files (page 41).

> **TIP**
>
> If the installation fails and you cannot identify the reason by viewing the log files, please contact our customer support.

Log Files

The installation log files are located in the *logs* subdirectory of the installer package (the directory to which you have expanded the PharmaSuite installer package).

To identify what caused the issue, proceed as follows:

1. Open the *InstallationProgress.log* file and find the step that caused the installer to abort.
2. Then open the log file of that step and read the reason why the installation failed.

Setting up the Live Data Infrastructure for PharmaSuite

This section contains general information about the system setup when PharmaSuite reads, writes, and browses process data values from automation devices on a factory floor network by means of FactoryTalk Live Data, a part of the FactoryTalk Services Platform (FTSP) from Rockwell Automation.

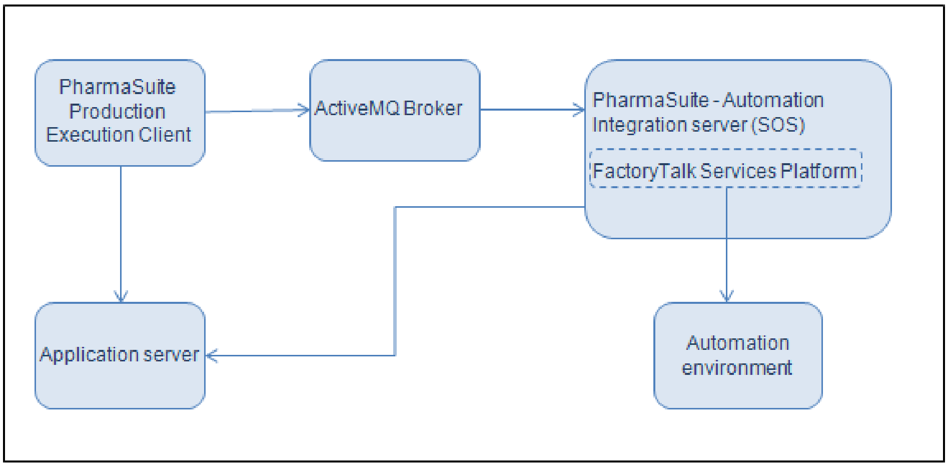

PharmaSuite Production Execution Client
ActiveMQ Broker
PharmaSuite - Automation Integration server (SOS)
FactoryTalk Services Platform
Application server
Automation environment

Figure 8: Overview of applied components

TIP

For more information, refer to FactoryTalk Help [B2 (page 72)], the online help of FactoryTalk Services Platform.

The PharmaSuite Automation Integration server is also used as Historian Integration server, see section "Setting up the Historian Infrastructure for PharmaSuite" (page 63).

Step 1: Network Considerations

Define a network node to be used by the Windows-based PharmaSuite Automation Integration server with access to all of the following systems:

- FactoryTalk ProductionCentre with JBoss as application server
- PharmaSuite Production Execution Clients
  For the ports used by the ActiveMQ JMS message broker, see "Setting up the ActiveMQ JMS Message Broker" (page 19).
- Data servers (e.g. OPC servers) (This includes access to the automation network through an installed and configured FactoryTalk Services Platform.)

Step 2: Topology Considerations

You may consider the following alternatives for various reasons:

- Performance, install the FactoryTalk Services Platform on the same computer as the Production Execution Client of PharmaSuite for direct access between Production Execution Client and the Live Data server (without JMS via the Automation Integration server).
- Scalability, install several Automation Integration servers to achieve load balancing (see "Step 11: Optional: Installing an Additional PharmaSuite Automation Integration Server" (page 57)).
- High availability, install several redundant Automation Integration servers to provide failover support (see "Step 10: Optional: Redundant PharmaSuite Automation Integration Servers for High Availability" (page 53)).
- Network segmentation, allow access to separated automation networks.
  Example: Each of your n different production lines is configured as a separate network and each production line has its own set of data servers. As a consequence you may need N Automation Integration servers.

For other configuration scenarios, see section "Automation Integration Configuration Scenarios" (page 59).

Step 3: FactoryTalk Services Platform Installation

Please refer to section "FactoryTalk Compatibility" in "FactoryTalk ProductionCentre Release 10.4 Supported Platforms Guide" [A1], for the version of the FactoryTalk Services Platform (FTSP) compatible with the current FactoryTalk ProductionCentre version.
FactoryTalk Services Platform is available from the Rockwell Automation Download Site.

Install FactoryTalk Services Platform to make the LiveData infrastructure available.

TIP

Installation of FactoryTalk Services Platform is not necessary, if you have already an automation infrastructure based on LiveData available and if one of the computers on which FactoryTalk Services Platform is installed shall be used to host the Automation Integration server.

When you install FactoryTalk Services Platform make sure that you are logged-on with an Administrator account into the console of the computer where the installation will take place (Remote Desktop connections cannot be used to successfully install FactoryTalk Services Platform).

If you plan to use data servers or the LiveData directory on other computers, we recommend to have all these computers in a common domain and all users administered with ActiveDirectory (setup in a workgroup environment is not recommended).

For details, see FactoryTalk Help [B2 (page 72)], the online help of FactoryTalk Administration Console.
The FactoryTalk Services Platform installer also installs FactoryTalk Administration Console, which is required in "Step 6: FactoryTalk Services Platform Configuration" (page 46) and Rockwell Software Data Client, which is required in "Step 7: Installation Verification" (page 47).

Step 4: Optional: OPC Simulator

We recommend to install and set up an OPC simulation server for training and testing purposes.

Step 5: Data Server

Install and set up a data server (e.g. RSLinx [A7] (page 71)). Define some test tags to verify the connection and the communication between the data server and PharmaSuite.

Step 6: FactoryTalk Services Platform Configuration

It is necessary to configure a network directory even for standalone installations.

Optional task:

- If you have set up another server to host the FactoryTalk directory or if you already have a FactoryTalk directory infrastructure, specify that directory. For this purpose, proceed as follows:

    1. In FactoryTalk Directory Server Location Utility, browse for the computer hosting the Network Directory Server and log on with one of the configured FactoryTalk users.
    2. Select the **Remote computer** option and browse for the server on which the FactoryTalk directory has been set up.
    3. When prompted, log on to the server with the credentials of a configured FactoryTalk user.

For details, see FactoryTalk Help [B2 (page 72)], the online help of FactoryTalk Directory Server Location Utility.

To configure the connection to the data servers with FactoryTalk Services Platform, proceed as follows:

1. In FactoryTalk Administration Console, select the **Network** option for the FactoryTalk directory.
2. To access the **Security Network Properties**, in the **Explorer** window, navigate to **System | Policies | System Policies**, right-click **Security Policy**, and define the **Computer policy settings** according to your environment.
3. To create FactoryTalk users, in the **Explorer** window, navigate to **System | Users and Groups**, and create a new user to be used to start the Shop Operations Server service.
   A FactoryTalk user can be created as an internal user or from a Windows-linked user.
4. Add the newly created user to the **Administrator** user group.
5. In the **Explorer** Window, right-click **Network** to create the hierarchy of Application, Area, and Data servers to connect the data servers.
   The default path (*RNA://$Global/LiveDataArea*) configured by PharmaSuite uses the **LiveDataArea** application without an area.

For details, see FactoryTalk Help [B2 (page 72)], the online help of FactoryTalk Administration Console.

TIP

Depending on your configuration it may be necessary to apply additional settings, e.g. configure DCOM security settings at the operating system level for each LiveData client computer and the data servers.

Step 7: Installation Verification

To verify the installation and configuration of the FactoryTalk Services Platform, proceed as follows:

1. On the computer on which FactoryTalk Services Platform was installed, start Rockwell Software Data Client (Live Data Test Client) and select the **Network** option for the FactoryTalk directory.
   For details, see Live Data Test Client Help [B3] (page 72).
2. In the **Initial Connection** dialog box, navigate to your FactoryTalk Area or FactoryTalk Application containing your OPC simulator or data server.
3. In the **Create Group** dialog box, click **OK** to create a group for your data items.
4. In the **Add Item** dialog box, navigate to your tags, click **Add Branch** and **OK**.
5. The Live Data Test Client window displays the added data items from the created group.
   If you can change the items by means of the **Write Items | Async Write** context menu item, the configuration of FactoryTalk Services Platform was successful.
6. On the computer on which FactoryTalk Services Platform was installed and the Automation Integration server is to be installed, start Process Designer.
7. Create a **LiveData TagSet Definition** object.
   For details, see section "LiveData TagSet Definitions" in "Process Designer Online Help" [B1] (page 72).
8. Use the FactoryTalk ProductionCentre **Live Data Browser** to browse to the Area or Application containing your OPC simulator or data server. If you see the already created tags and can create LiveData tags from them, then the FactoryTalk ProductionCentre connectivity to FactoryTalk Services Platform is verified.

Step 8: Installing the PharmaSuite Automation Integration Server for Live Data

PharmaSuite offers an interface for synchronous service calls to read, write, monitor, or verify tags. For this purpose, it internally uses JMS (ActiveMQ) for the communication between the PharmaSuite Automation Integration server and PharmaSuite clients. The requests are performed on FactoryTalk Live Data to return the result to the client.

To install the Automation Integration server observe the instructions in section "Setting up Shop Operations Servers" (page 23) and assign the **PharmaSuite_AI_Server** event sheet.

The user rights to access FactoryTalk Live Data can be configured as follows:

- By default, the event sheet runs as a Windows service under the system account.
- Alternative 1: Run the service with a Windows user account that has access to FactoryTalk Live Data. For this purpose, in the Windows service management, adapt the **PharmaSuite AI server** service (**Log On** tab).
- Alternative 2: Configure the Automation Integration server to use a specific Live Data user. For this purpose, adapt the **Equipment/LiveDataUsername** and **Equipment/LiveDataPassword** configuration keys (see chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71)).

IMPORTANT

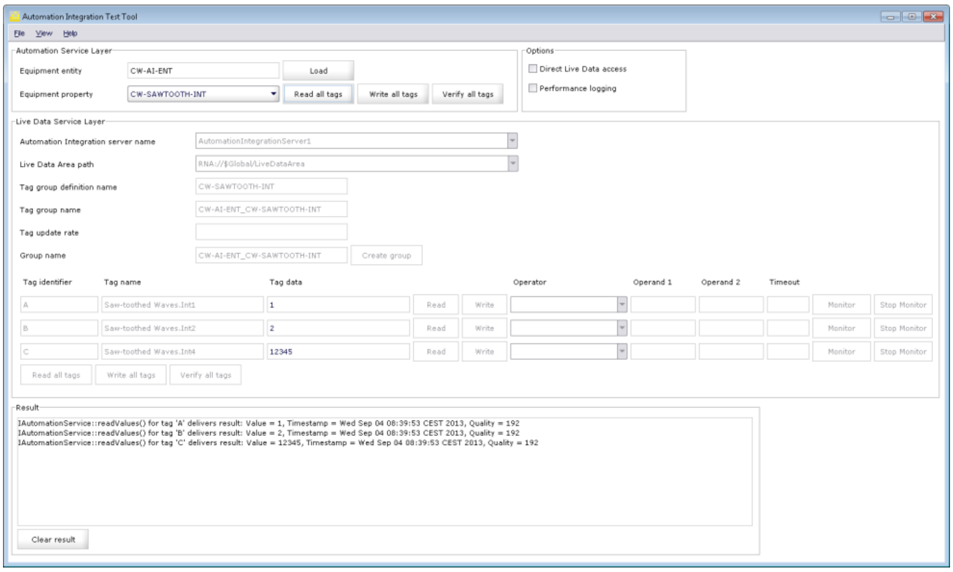
Make sure that the date and time settings are synchronized between the machines that are running the PharmaSuite clients, the ActiveMQ JMS message broker, and the Automation Integration server. For details, see section "Ensuring Synchronized Date and Time Settings" (page 35) in chapter "Installing PharmaSuite" (page 3).

Step 9: Verifying the Access from Production Execution Client to Automation Integration Server

After you have installed and configured the required components, we recommend to verify the access to the Automation Integration server.

The **Automation Integration Test** tool supports you with functions for checking the Live Data infrastructure (e.g. communication from a data server to PharmaSuite and vice versa).

- High level functions to verify the interface used by PharmaSuite phase building blocks (page 50)
  Read, write, and verify tags (value, timestamp, quality, and errors) of the data based on the S88 equipment-related master data (equipment entity and property), which are maintained in Data Manager.
- Low level functions to verify and expose the pure interface without accessing S88 equipment-related master data (page 50)
  Read, write, monitor, and verify tags (value, timestamp, quality, and errors).



Automation Service Layer

Equipment entity: CW-AI-ENT

Equipment property: CW-SAWTOOTH-INT

Options

□ Direct Live Data access

□ Performance logging

Live Data Service Layer

Automation Integration server name: AutomationIntegrationServer1

Live Data Area path: RNA://$Global/LiveDataArea

Tag group definition name: CW-SAWTOOTH-INT

Tag group name: CW-AI-ENT_CW-SAWTOOTH-INT

Tag update rate:

Group name: CW-AI-ENT_CW-SAWTOOTH-INT

Tag identifier	Tag name	Tag data	Operator	Operand 1	Operand 2	Timeout
A	Saw-toothed Waves.Int1	1				
B	Saw-toothed Waves.Int2	2				
C	Saw-toothed Waves.Int4	12345				

Result

```
IAutomationService::readValues() for tag 'A' delivers result: Value = 1, Timestamp = Wed Sep 04 08:39:53 CEST 2013, Quality = 192
IAutomationService::readValues() for tag 'B' delivers result: Value = 2, Timestamp = Wed Sep 04 08:39:53 CEST 2013, Quality = 192
IAutomationService::readValues() for tag 'C' delivers result: Value = 12345, Timestamp = Wed Sep 04 08:39:53 CEST 2013, Quality = 192
```

Figure 9: Verify access to Automation Integration server

TIP

Even though the **FlexibleTagDefinition** data type may have more than three tag definitions, the tool only supports the first three tag definitions.

To use the tool, run the **mes_AutomationIntegrationTestTool** form to start the **Automation Integration Access Verification** tool.

High Level Functions

The following high level API functions are available using the Automation service layer:

- Optional: Show profiling information
  Enable the **Performance logging** option to display how much time is used for the corresponding API call in the **Result** panel.
- Load an equipment entity
  Type an entity identifier and click **Load**. The tool displays the available data of the equipment entity and the associated Live Data-related data as read-only (except for the tag data, which is always editable).
- Properties of a loaded equipment entity
  The option list provides all of the entity's properties of the **Automation** usage type sorted by name. The first entry is selected automatically, others can be selected manually.
  The tool updates the display of the property type-related data and of the associated Live Data-related data.
- Read all tags
  To read all of the tags of the selected property, click the **Read all tags** button in the **Automation Service Layer** panel.
  The system displays the tag data in the corresponding text boxes and the API call in the **Result** panel.
- Write all tags
  To write all of the tags of the selected property, type the data in the corresponding text box and click the **Write all tags** button in the **Automation Service Layer** panel.
  The system displays the API call in the **Result** panel.
- Verify all tags
  To verify all of the tags of the selected property, click the **Verify all tags** button in the **Automation Service Layer** panel.
  The system displays the API call in the **Result** panel.

Low Level Functions

The following low level API functions are available using the Live Data service layer:

- Optional: Show profiling information
  Enable the **Performance logging** option to display how much time is used for the corresponding API call in the **Result** panel.
- Optional: Load an equipment entity
  Type an entity identifier and click **Load**. The tool displays the available data of the equipment entity and the associated Live Data-related data as read-only (except for the tag data, which is always editable).

- Optional: Properties of a loaded equipment entity
  The option list provides all of the entity's properties of the **Automation** usage type sorted by name. The first entry is selected automatically, others can be selected manually.
  The tool updates the display of the property type-related data and of the associated Live Data-related data.

- Write access to Live Data-related data
  Enable the **Direct Live Data access** option to make all text boxes in the **Live Data Service Layer** panel editable.

- Automation Integration server name
  The default value is read from the **Default configuration** application (see **Equipment/DefaultAutomationIntegrationServerName** configuration key, chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71).)
  The option list provides other available Automation Integration servers.

  - *LiveDataAccessFileMock* is useful for local testing without a real FactoryTalk Services Platform LiveData connection. The corresponding mock file of a Live Data group has a unique name and is located in *c:\Users\<user>\AppData\Local\Temp*. The file is created, unless it already exists.

  - *LocalLiveDataAccess* is useful when FactoryTalk Services Platform is installed and the **Automation Integration Test** tool is running on the same computer (see section "Automation Integration Configuration Scenarios" (page 59)).

  You can also type a server name in the text box.

- Live Data Area path
  The default value is read from the **Default configuration** application (see **Equipment/DefaultLiveDataAreaPath** configuration key, chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71).)
  You can also type a server path in the text box.

- Tag group definition name
  Type the name of an existing tag group definition. By default, the name corresponds to the identifier of the equipment property.

- Tag group name
  Type the name of an existing tag group. By default, the name is specified as follows: **<equipment entity identifier>_<equipment property identifier>**.

- Tag update rate
  Type the tag update rate for the group, otherwise the default tag update rate is used.

- Group name
  Type the name of a group to be used during its creation. The group is used when accessing tags and is created on the server. By default, the name is the same as the tag group name (**<equipment entity identifier>_<equipment property identifier>**).
- Create a Live Data group
  Load an equipment entity or type a tag set definition name, tag set name, and group name, then click the **Create group** button.
  This is only necessary **once** for any given tag group definition.
- Tag identifier 1..3
  Identifiers of the tags of the current tag group.
- Tag name 1..3
  Tag names of the related tag identifier. The name is read-only.
- Tag data 1..3
  Data of the related tag identifier.
- Only for monitoring: Operator
  The option list provides comparison operators to be used when tag data is monitored.
- Only for monitoring: Operand 1
  First operand of the comparison operation.
- Only for monitoring: Operand 2
  Second operand of the comparison operation.
- Only for monitoring: Timeout
  Timeout in milliseconds to be used when tag data is monitored.
- Read tags
  To read one or all tags, click the **Read** or **Read all tags** button in the **Live Data Service Layer** panel.
  The system displays the tag data in the corresponding text boxes and the API call in the **Result** panel.
- Write tags
  To write one or all tags, type the data in the corresponding text box and click the **Write** or **Write all tags** button in the **Live Data Service Layer** panel.
  The system displays the API call in the **Result** panel.
- Only for monitoring: Monitor tags
  To monitor the change of tag data for a given condition, click the **Monitor** button in the **Live Data Service Layer** panel. The condition is defined with the **Operator**, **Operand 1**, **Operand 2**, and **Timeout** parameters.
  An API call for monitoring is scheduled in a separate thread, so monitoring of multiple tags is possible.

- Only for monitoring: Stop monitoring tags
  To stop monitoring of tag data, click the **Stop monitor** button in the **Live Data Service Layer** panel.
  An API call to stop monitoring is executed immediately.
- Verify tags
  To verify all of the tags, click the **Verify all tags** button in the **Live Data Service Layer** panel.
  The system displays the API call in the **Result** panel.

> TIP
>
> The "all" buttons only apply to tags for which a name has been defined.
>
> A typical use case for the low level functions is to load an equipment entity and an equipment property first to get suitable default values for some low level parameters based on the loaded equipment data. Then, switch on the **Direct Live Data access** option to enable editing of the parameters.

Step 10: Optional: Redundant PharmaSuite Automation Integration Servers for High Availability

This section applies to the Live Data and the Historian infrastructure (page 63).

The high availability scenario requires at least several computers:

1. The **first** Automation Integration server computer where the primary Automation Integration server is running.
2. The **second** Automation Integration server computer where the secondary Automation Integration server is running.
3. Only for Live Data infrastructure:
   The **third** one is the one where the LiveData directory is hosted. This is the recommended setup.
4. (Optional) The computer(s) where
   - the data servers configured in the above LiveData directory or
   - the Historian access servers and the Historian servers

   are running.
   This can be the same computer as above.

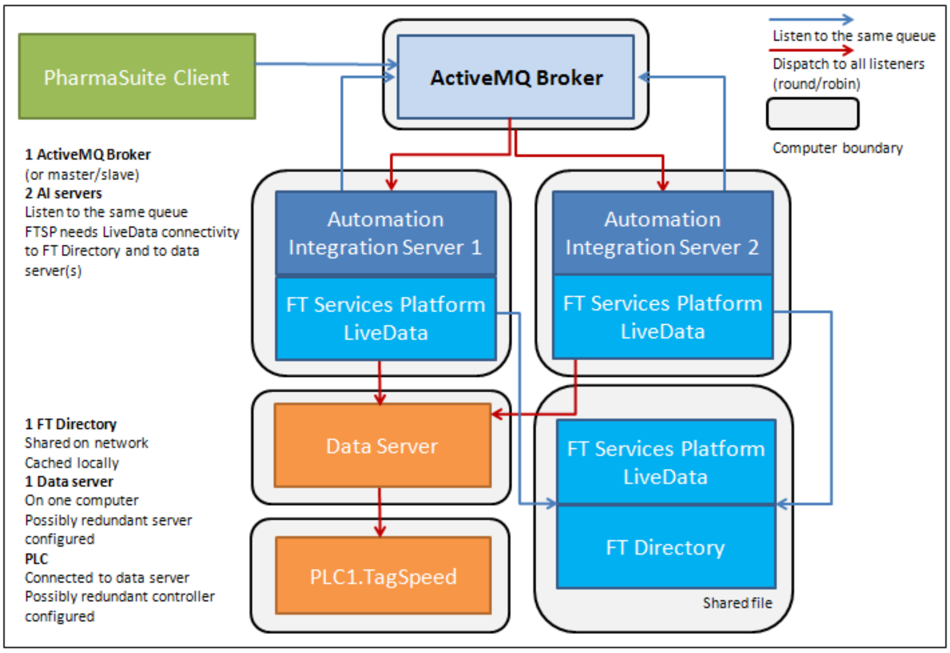

PharmaSuite Client
ActiveMQ Broker
Listen to the same queue
Dispatch to all listeners (round/robin)
Computer boundary
1 ActiveMQ Broker
(or master/slave)
2 AI servers
Listen to the same queue
FTSP needs LiveData connectivity to FT Directory and to data server(s)
Automation Integration Server 1
Automation Integration Server 2
FT Services Platform LiveData
FT Services Platform LiveData
1 FT Directory
Shared on network
Cached locally
1 Data server
On one computer
Possibly redundant server configured
PLC
Connected to data server
Possibly redundant controller configured
Data Server
FT Services Platform LiveData
FT Directory
PLC1.TagSpeed
Shared file

Figure 10: Example: High availability (with redundant hardware for Automation Integration servers)

An alternative setup uses only one server computer running two Automation Integration servers. This provides protection from software failures of these servers only, not from hardware or system failures of the computer.

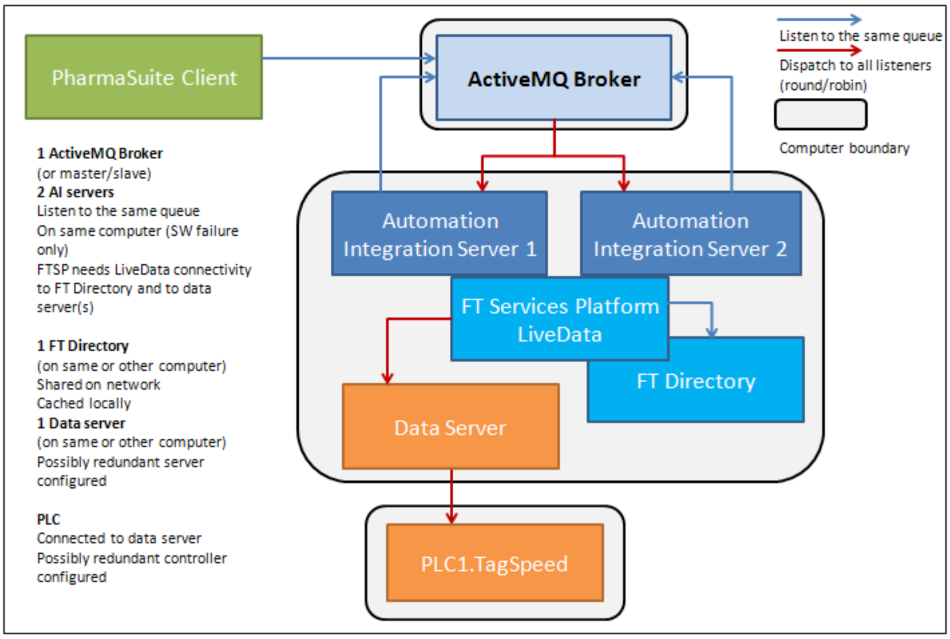

PharmaSuite Client
ActiveMQ Broker
Listen to the same queue
Dispatch to all listeners (round/robin)
Computer boundary
1 ActiveMQ Broker
(or master/slave)
2 AI servers
Listen to the same queue
On same computer (SW failure only)
FTSP needs LiveData connectivity to FT Directory and to data server(s)
1 FT Directory
(on same or other computer)
Shared on network
Cached locally
1 Data server
(on same or other computer)
Possibly redundant server configured
PLC
Connected to data server
Possibly redundant controller configured
Automation Integration Server 1
Automation Integration Server 2
FT Services Platform LiveData
FT Directory
Data Server
PLC1.TagSpeed

Figure 11: Example: High availability (with redundant software for Automation Integration servers)

Only for Live Data infrastructure:

- On all the computers, FactoryTalk Services Platform has to be installed and set up in a way that on both Automation Integration server computers, the tags from the configured data servers can be accessed.
- FactoryTalk Services Platform is responsible for the high availability of the LiveData directory (in case the third computer fails or is not accessible). Additionally, FactoryTalk Services Platform can configure redundancy for the data servers itself. For details, see FactoryTalk Help [B2] (page 72).

Only for Historian infrastructure:

- On all the computers, the PI JBDC driver has to be installed (page 65).

Installing a Redundant PharmaSuite Automation Integration Server

This section applies to the Live Data and the Historian infrastructure (page 63).

To install a redundant Automation Integration server to support the high availability scenario, perform the following steps:

1. Make sure that the first Automation Integration server has been installed (see "Step 8: Installing the PharmaSuite Automation Integration Server for Live Data" (page 48) or "Step 6: Installing the PharmaSuite Automation Integration Server for Historian" (page 67)).
2. On the second Automation Integration server computer
   - Only for Live Data infrastructure:
     Repeat "Step 3: FactoryTalk Services Platform Installation" (page 45) and "Step 6: FactoryTalk Services Platform Configuration" (page 46).
     - Step 6: FactoryTalk Services Platform Configuration
       Make sure to use the same FactoryTalk directory as for the first computer.
   - Only for Historian infrastructure:
     Repeat "Step 3: PI JDBC Driver Installation" (page 65) and "Step 4: FactoryTalk ProductionCentre Configuration" (page 65).
3. The Shop Operations Server for the second Automation Integration server uses the same **Application** object as the first Automation Integration server and refers to the same configuration key (e.g. **Equipment/DefaultAutomationIntegrationServerName = AutomationIntegrationServer1**, **Equipment/DefaultHistorianAIServerName = AutomationIntegrationServer1**).
   Only for Live Data infrastructure:
   The Shop Operations Server runs under the same user.

4. In the following, *<path_to_service>* is used as a placeholder for the location of the *PharmaSuite_EBR_Server* server (e.g. *C:\Rockwell\PharmaSuite\installation\services\*).
   If you install the redundant Automation Integration server on a computer where the PharmaSuite Installer has not been run, copy the *<path_to_service>\PharmaSuite_AI_Server* directory from the computer where PharmaSuite and the first Automation Integration server has been installed to the same path on the computer for the second Automation Integration server.
   Open *<path_to_service>\PharmaSuite_AI_Server3\conf\wrapper.conf* and replace each uncommented occurrence of `.../j2sdk/...` with the path where the required Java SDK version has been installed.
   If JBoss and ActiveMQ are not running on the server, remove the corresponding dependency (`wrapper.ntservice.dependency.*`).

5. Set up the Shop Operations Server authentication.
   In *{SOSInstallDir}\bin\ShopOperationsServer.xml* enter the user name and password of the user.
   Navigate to *<path_to_service>\PharmaSuite_AI_Server\bin\* and run *InstallApp-NT.bat* as administrator.
   Open **Services**, navigate to the **PharmaSuite AI server** service, and start the service.
   Open *<path_to_service>\PharmaSuite_AI_Server\logs\wrapper.log\* and review the last message. It should be similar to the sample log shown below.

   ```
   [WrapperSimpleAppMain]  WARN
   AutomationIntegrationServerActivity:35 - Started
   AutomationIntegrationServer with name
   AutomationIntegrationServer1 with messaging address
   failover:(tcp://<activeMQServer>:61646)?randomize=false&time
   out=1000
   ```

(Optional) Only required for the scenario with two Automation Integration servers on the same computer:

- Choose a different target directory for the copy e.g. *<path_to_service>\PharmaSuite_AI_Server2*
- In *wrapper.conf*, replace each occurrence of `AutomationIntegrationServer1` with, for example, `AutomationIntegrationServer2`.
- In *{SOSInstallDir}\bin\ShopOperationsServer.xml*, specify a free port number, for example, replace `jetty-port="8086"` with `jetty-port="8087"` (or another free port).

Step 11: Optional: Installing an Additional PharmaSuite Automation Integration Server

This section applies to the Live Data and the Historian infrastructure (page 63).

The figure below illustrates the scenario where an additional Automation Integration server is required since different Automation Integration servers are addressed by different equipment properties.

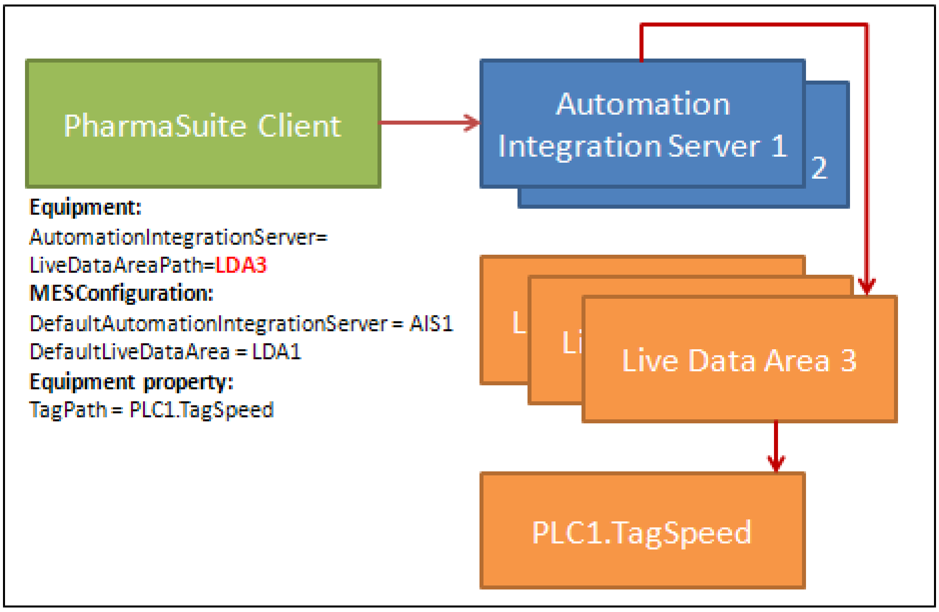


Figure 12: Example: Specific Live Data Area per equipment

To set up your configuration according to the scenario, perform the following steps:

1. In Process Designer, create a new **Application** object for the new Automation Integration server configuration, based on the **DefaultConfiguration**.
   PharmaSuite provides the PS Administration application to administer database objects such as access privileges, lists, applications, users, or user groups, see "Implementation Guide PS Administration" [A10] (page 71).
   Using Process Designer for these tasks is possible, but does not provide the full functional scope required for PharmaSuite. For information on the administration of database objects with Process Designer in this case, please refer to chapter "Managing Configurations" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71).
   Example: **AIServer2Config**
2. For Live Data infrastructure:
   Set the value of the **Equipment/DefaultAutomationIntegrationServerName** configuration key as follows: **AutomationIntegrationServer2**

3. For Historian infrastructure:
   Set the value of the **Equipment/DefaultHistorianAIServerName** configuration key as follows: **AutomationIntegrationServer2**
4. Create a new user with **AIServer2User** as name, for example.
5. Bind **AIServer2Config** to the **AIServer2User** user.
   In the **User** object, set the **BootstrapApp** parameter to the **AIServer2Config** application.
6. Install the new Automation Integration server according to the instructions in section "Setting up Shop Operations Servers" (page 23) and assign the **PharmaSuite_AI_Server** event sheet.
7. Make the Shop Operations Server authenticate as **AIServer2User** user.
   - In *{SOSInstallDir}\bin\ShopOperationsServer.xml* enter the user name and password of the **AIServer2User** user.

Configuration Keys

The following configuration keys are available for setting up a Live Data infrastructure:

- Equipment/AIServerMessagingTimeoutInSeconds
- Equipment/DefaultAutomationIntegrationServerName
- Equipment/DefaultLiveDataAreaPath
- Equipment/DefaultTagUpdateRateInMilliseconds
- Equipment/LiveDataPassword
- Equipment/LiveDataUsername
- Equipment/TagQualityGoodList

For details, chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71).

Automation Integration Configuration Scenarios

The following configuration scenarios illustrate specific use cases:

TIP

To improve legibility, the Automation Integration server name and path to the Live Data Area are abbreviated.

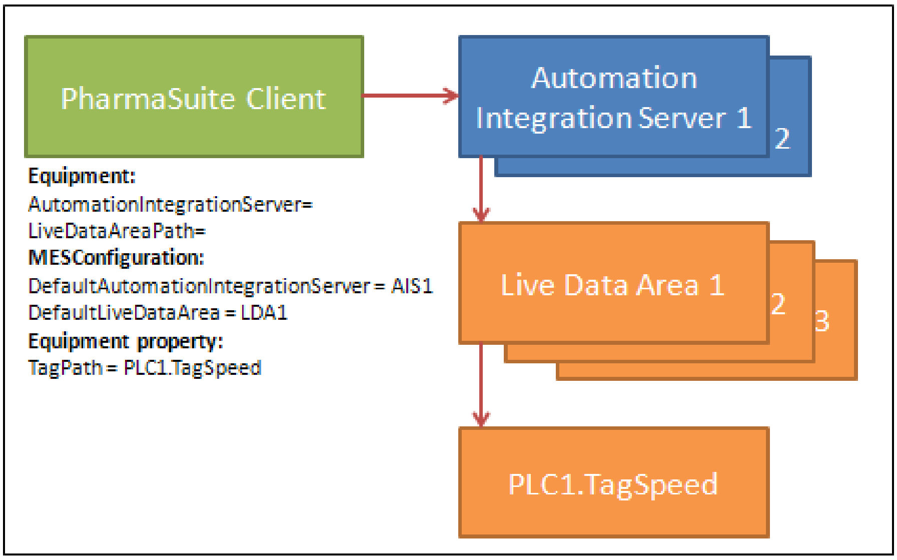


Figure 13: Example: Normal access

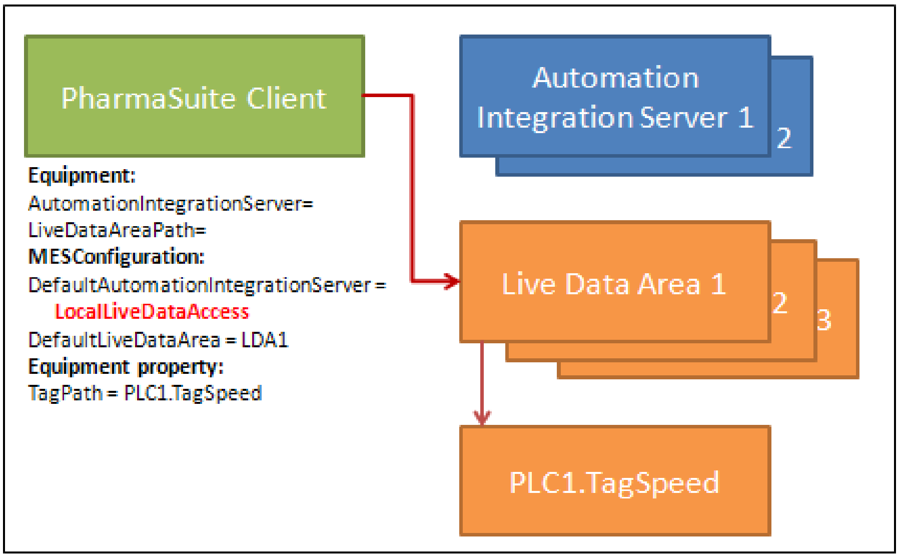


Figure 14: Example: Direct access for performance reasons

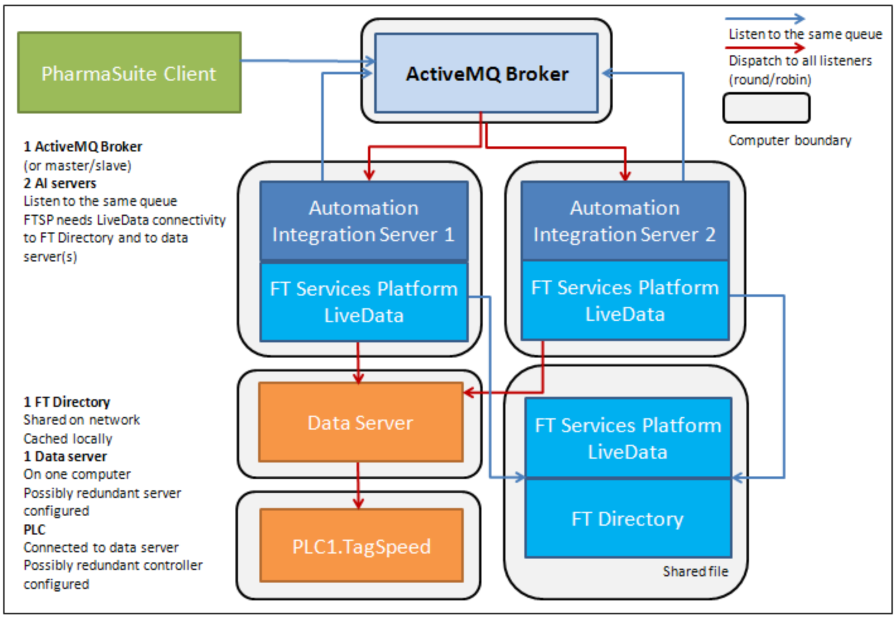

PharmaSuite Client
ActiveMQ Broker
Listen to the same queue
Dispatch to all listeners (round/robin)
Computer boundary
1 ActiveMQ Broker
(or master/slave)
2 AI servers
Listen to the same queue
FTSP needs LiveData connectivity to FT Directory and to data server(s)
Automation Integration Server 1
FT Services Platform LiveData
Automation Integration Server 2
FT Services Platform LiveData
1 FT Directory
Shared on network
Cached locally
1 Data server
On one computer
Possibly redundant server configured
PLC
Connected to data server
Possibly redundant controller configured
Data Server
PLC1.TagSpeed
FT Services Platform LiveData
FT Directory
Shared file

Figure 15: Example: High availability (with redundant hardware for Automation Integration servers)

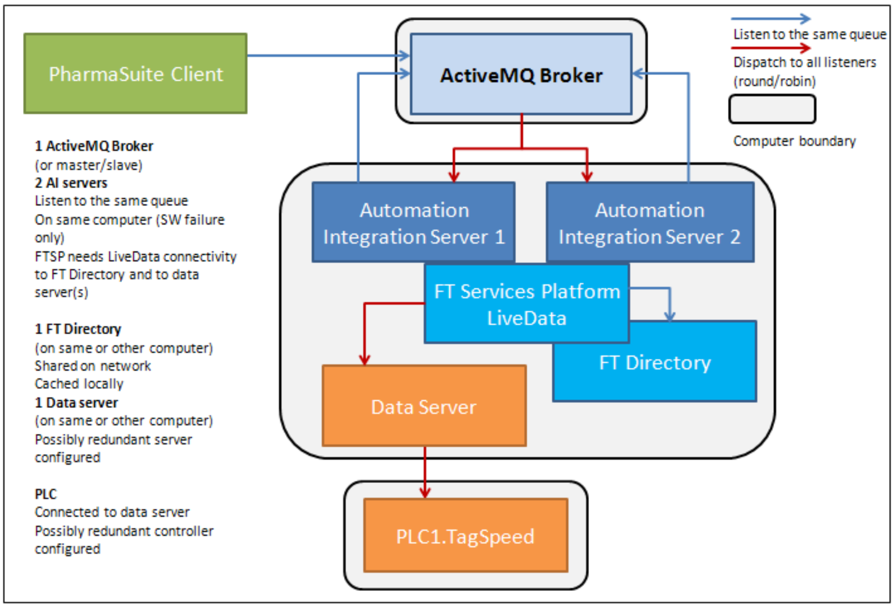

PharmaSuite Client
ActiveMQ Broker
Listen to the same queue
Dispatch to all listeners (round/robin)
Computer boundary
1 ActiveMQ Broker
(or master/slave)
2 AI servers
Listen to the same queue
On same computer (SW failure only)
FTSP needs LiveData connectivity to FT Directory and to data server(s)
1 FT Directory
(on same or other computer)
Shared on network
Cached locally
1 Data server
(on same or other computer)
Possibly redundant server configured
PLC
Connected to data server
Possibly redundant controller configured
Automation Integration Server 1
Automation Integration Server 2
FT Services Platform LiveData
FT Directory
Data Server
PLC1.TagSpeed

Figure 16: Example: High availability (with redundant software for Automation Integration servers)

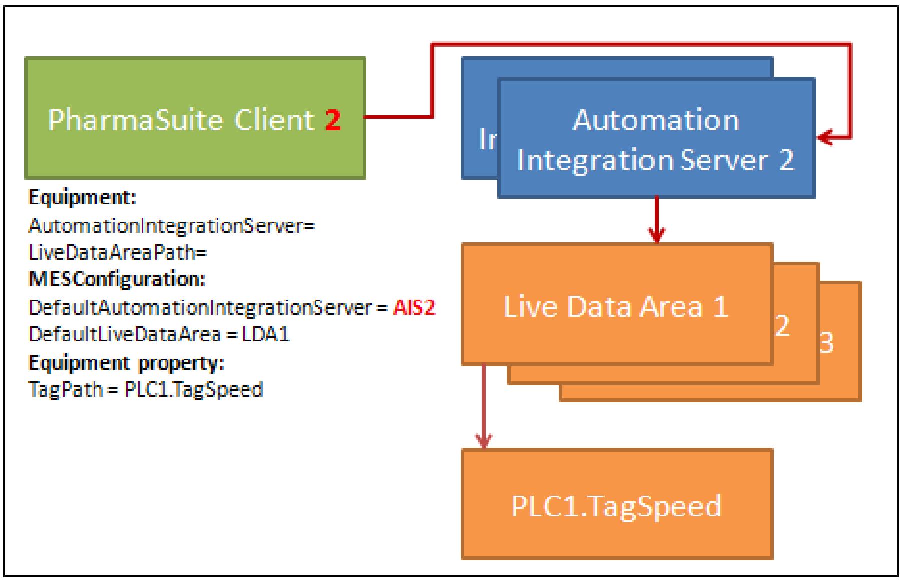


Figure 17: Example: Load balancing by different client configurations

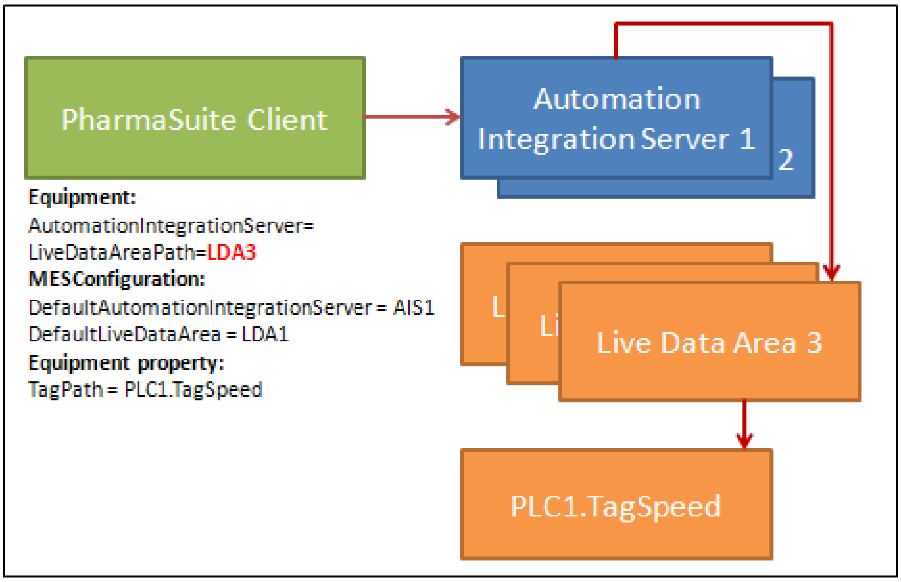
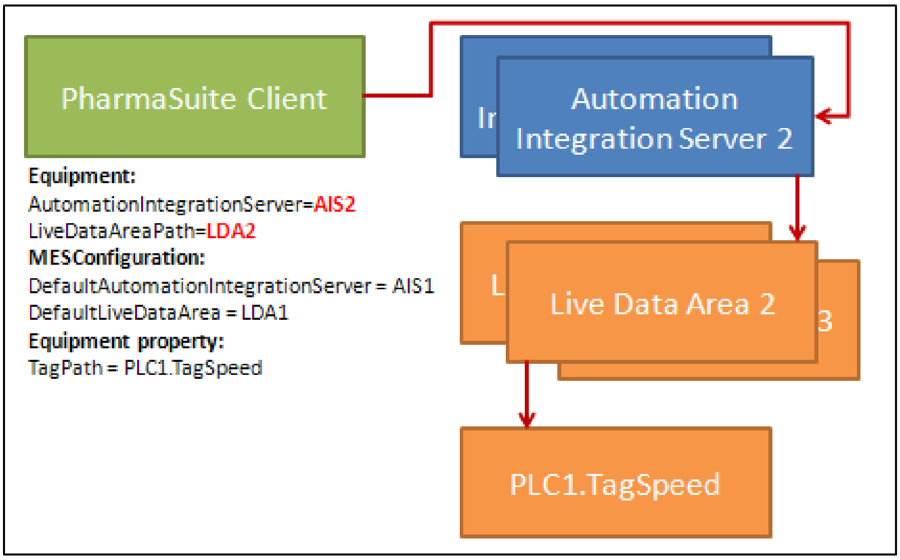
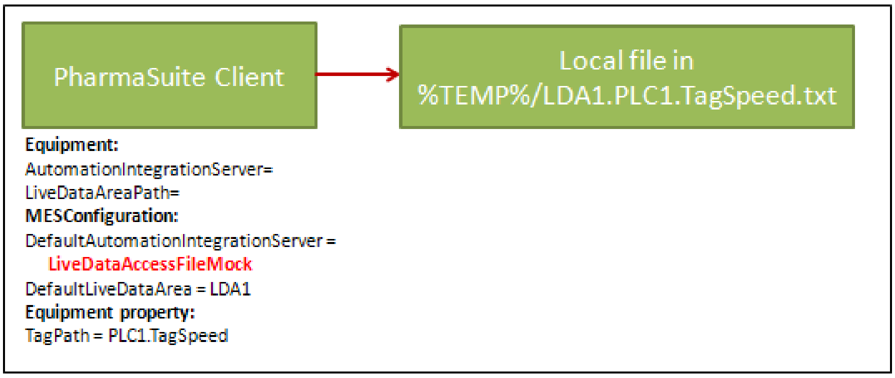


Figure 18: Example: Specific Live Data Area per equipment



Equipment:
AutomationIntegrationServer=AIS2
LiveDataAreaPath=LDA2
MESConfiguration:
DefaultAutomationIntegrationServer = AIS1
DefaultLiveDataArea = LDA1
Equipment property:
TagPath = PLC1.TagSpeed

Figure 19: Example: Segmented automation networks



Equipment:
AutomationIntegrationServer=
LiveDataAreaPath=
MESConfiguration:
DefaultAutomationIntegrationServer =
LiveDataAccessFileMock
DefaultLiveDataArea = LDA1
Equipment property:
TagPath = PLC1.TagSpeed

Figure 20: Example: Development-focused test

Setting up the Historian Infrastructure for PharmaSuite

This section contains general information about the system setup when PharmaSuite reads and browses historical data values from a Historian server on a factory floor network by means of FactoryTalk Historian SE from Rockwell Automation.
Typically, the Historian environment is accessed indirectly by the PharmaSuite Historian Integration server (i.e. the Automation Integration server).

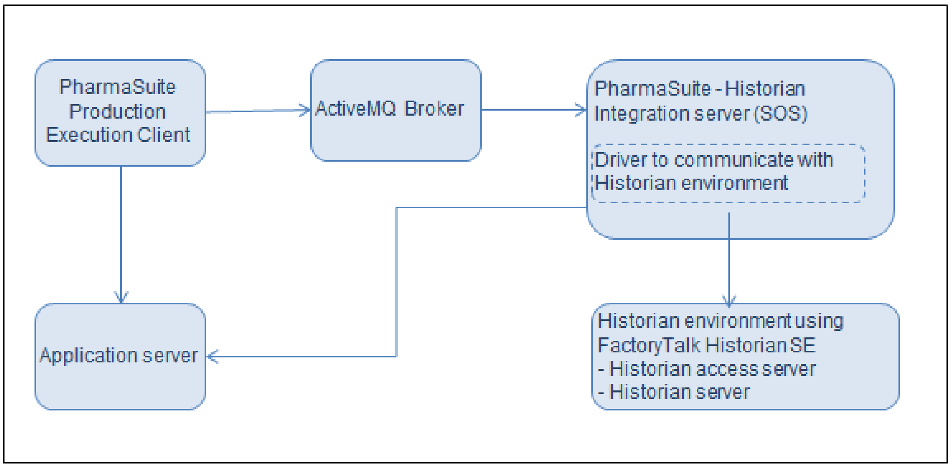

PharmaSuite Production Execution Client
ActiveMQ Broker
PharmaSuite - Historian Integration server (SOS)
Driver to communicate with Historian environment
Application server
Historian environment using FactoryTalk Historian SE
- Historian access server
- Historian server

Figure 21: Overview of applied components

TIP

The instructions assume that FactoryTalk Historian SE (Historian server, Historian access server = PI SQL Data Access Server) is already installed, set up, and running. For the supported version, see "FactoryTalk ProductionCentre Release 10.4 Supported Platforms Guide" [A1] (page 71).
The Historian access server hosts interfaces for the communication with the Historian server. The Historian server is a collection of modules responsible for processing events for each configured data point, configuring them, and archiving the collected data.

Step 1: Network Considerations

Define a network node to be used by the Windows-based PharmaSuite Automation Integration server with access to all of the following systems:

- FactoryTalk ProductionCentre with JBoss as application server
- PharmaSuite Production Execution Clients
  For the ports used by the ActiveMQ JMS message broker, see "Setting up the ActiveMQ JMS Message Broker" (page 19).
- Historian access server (e.g. Data Access Server for PI SQL Clients)
  Direct access to the Historian server is not needed since it is only used indirectly via the Historian access server.

Step 2: Topology Considerations

You may consider the following alternatives for various reasons:

- Performance, install PI JDBC 2010 R3 on the same computer as the Production Execution Client of PharmaSuite for direct access between Production Execution Client and the Historian access server (without JMS via the Automation Integration server).
- Scalability, install several Automation Integration servers to achieve load balancing (see "Step 11: Optional: Installing an Additional PharmaSuite Automation Integration Server" (page 57) of the Live Data infrastructure).
- High availability, install several redundant Automation Integration servers to provide failover support (see "Step 10: Optional: Redundant PharmaSuite Automation Integration Servers for High Availability" (page 53) of the Live Data infrastructure).
- Network segmentation, allow access to separated Historian networks.
  Example: Each of your n different production lines is configured as a separate network and each production line has its own set of Historian access servers. As a consequence you may need N Automation Integration servers.

For other configuration scenarios, see section "Historian Integration Configuration Scenarios" (page 69).

Step 3: PI JBDC Driver Installation

Install the PI JDBC Driver to make the Historian SQL-based access available.

> **TIP**
>
> PI JDBC is compatible with the current version of PharmaSuite.
> The driver is available as part of FactoryTalk Historian Site Edition.

All machines accessing the Historian access server via JDBC need a local installation of the PI JDBC driver. This includes:

- the machine hosting the Automation Integration server used for central PI access and
- the machines running PharmaSuite Data Manager or Production Execution Client on an equipment instance whose Automation Integration server name is set to **LocalHistorianAccess**.

To install the PI JDBC driver, see FT Historian SE Installation and Configuration Guide [B5] (page 72).

Step 4: FactoryTalk ProductionCentre Configuration

Before you can use the *PIJDBCDriver.jar* driver .JAR file, you have to deploy the driver to PharmaSuite as a FactoryTalk ProductionCentre **Library** object. This has to be done only once.

In Process Designer, proceed as follows:

1. Create a new **Library** object with the following properties:
   - **Name**: PIJDBCDriver.jar
   - **Library Jar Path**: c:\Program Files\PIPC\JDBC\PIJDBCDriver.jar
2. For each Historian server, configure the credentials in the **Application** object.

Example:

The Historian server name is **FT-H-SE** and the following configuration keys are used in the **Equipment** class of the **Application** object.

Name	Value	Type
HistorianAccessPassword.FT-H-SE	-14,91,49,-34,-114,-117,79,-94,...	String
HistorianAccessUser.FT-H-SE	FT-H-SE\Administrator	String

You can store the password as clear text or encrypted. To encrypt the password, use the **mes_PasswordEncryptionForm** utility.

The names of the Historian access server (=PI Data Access server) and the actual Historian server (=PI server) are configured in PharmaSuite Data Manager as equipment entity attributes.

If the Automation Integration server name is set to **LocalHistorianAccess** instead of e.g. **AutomationIntegrationServer1**, the communication to the Historian access server will be performed directly from the PharmaSuite client using a local PI JDBC driver installation.

Step 5: Installation Verification (Part 1: JDBC Access)

To verify that the JDBC access is working, proceed as follows:

1. Make sure that local Java JRE is available on the machine where PI JDBC has been installed.
2. Open a command prompt, navigate to *c:\Program Files\PIPC\JDBC\Samples\getSnap\bin*
3. Type

```
java getSnap FT-H-SE FT-H-SE sin%
```

   where the three parameters are PI SQL DAS name, PI Server name, and PI tag name or tag name wildcard (SQL syntax).

4. The console output will be similar to:

```
com.osisoft.jdbc.Driver 1.2.2.0243
PI SQL DataAccessServer using PIOLEDB
PIOLEDB: 3.3.1.2

SINUSOID 30.510075
SINUSOIDU 0.08331537
```

For details, see OSIsoft PI JDBC 2017 Administrator Guide, page 15, [D3] (page 72).

Step 6: Installing the PharmaSuite Automation Integration Server for Historian

PharmaSuite offers an interface for synchronous service calls to read or verify tags from a Historian server. For this purpose, it internally uses JMS (ActiveMQ) for the communication between the PharmaSuite Automation Integration server and PharmaSuite clients. The requests are performed using JDBC against the PI Data Access server to return the result to the client.

To install the Automation Integration server observe the instructions in section "Setting up Shop Operations Servers" (page 23) and assign the **PharmaSuite_AI_Server** event sheet.

The user rights to access FactoryTalk Historian SE can be configured as follows:

- The JDBC connection is configured to use Integrated Security (SSPI) configuration for the PI Server login. For details, see OSIsoft, Configuring PI Server Security, [D3] (page 72).
- Configure PharmaSuite to use a specific user and password to login to each Historian access server. For this purpose, for each Historian data access server, add the **Equipment/HistorianAccessUser.<Historian access server name>** and **Equipment/HistorianAccessPassword.<Historian access server name>** configuration keys.
  See chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71).
  We recommend to use password encryption.

IMPORTANT

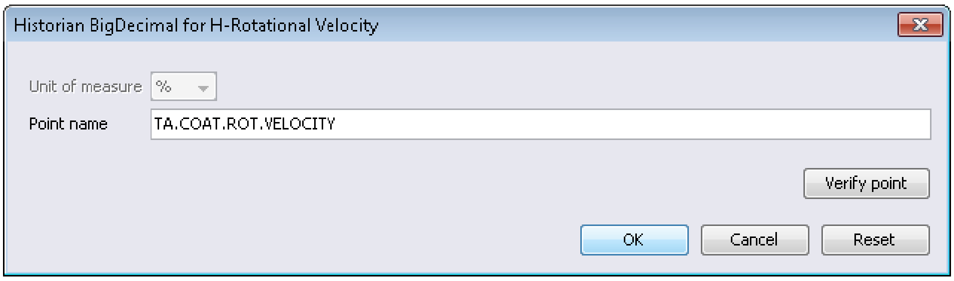
Make sure that the date and time settings are synchronized between the machines that are running the PharmaSuite clients, the ActiveMQ JMS message broker, and the Automation Integration server. For details, see section "Ensuring Synchronized Date and Time Settings" (page 35) in chapter "Installing PharmaSuite" (page 3).

Step 7: Installation Verification (Part 2: Library Object)

To verify the installed FactoryTalk ProductionCentre **Library** object, proceed as follows:

1. In PharmaSuite Data Manager, create a property type of the **Historian** usage type and **BigDecimal** data type.
2. Create an equipment entity and assign the newly created property type as process property in the **Process** tab.
   The verification assumes that the Historian attributes in the **Basic** tab are valid.
3. In the **Process** tab of the entity, open the Historian BigDecimal editor for the property.

4. Type an existing tag name of a numeric Historian tag (valid on the configured server, e.g. SINUSOID), click the **OK** button, and save the entity.
5. Open the Historian BigDecimal editor for the property again and click the **Verify point** button.
   An information message lets you know that the verification was successful (Point name verification was performed successfully.).



Historian BigDecimal for H-Rotational Velocity

Unit of measure: %

Point name: TA.COAT.ROT.VELOCITY

Figure 22: Verification of the FactoryTalk ProductionCentre Library object

Configuration Keys

The following configuration keys are available for setting up a Historian infrastructure:

- Equipment/HistorianAccessUser.<HistorianAccessServerName>
- Equipment/HistorianAccessPassword.<HistorianAccessServerName>
- Equipment/DefaultHistorianServerName
- Equipment/DefaultHistorianAccessServerName
- Equipment/DefaultHistorianAIServerName
- Equipment/PI_JDBC_DCA_OPTION
- Equipment/PI_JDBC_Property_LogFile
- Equipment/PI_JDBC_Property_LogConsole
- Equipment/PI_JDBC_Property_LogLevel
- Equipment/PI_JDBC_URL

For details, chapter "Configuration Keys of PharmaSuite" in Volume 4 of the "Technical Manual Configuration and Extension" [A6] (page 71).

Historian Integration Configuration Scenarios

The following configuration scenarios illustrate specific use cases:

> **TIP**
>
> To improve legibility, the Automation Integration server name and others are abbreviated.

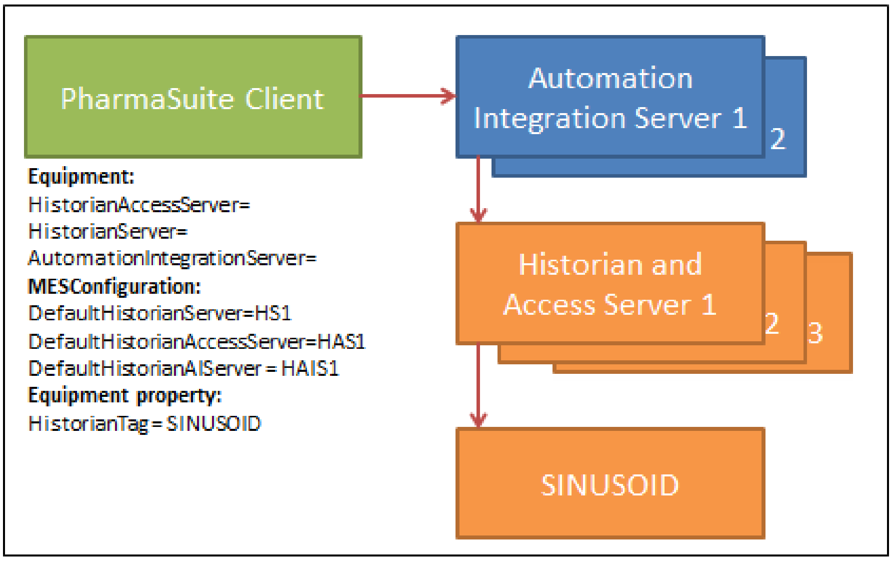

PharmaSuite Client
Automation Integration Server 1
2
Equipment:
HistorianAccessServer=
HistorianServer=
AutomationIntegrationServer=
MESConfiguration:
DefaultHistorianServer=HS1
DefaultHistorianAccessServer=HAS1
DefaultHistorianAIServer = HAIS1
Equipment property:
HistorianTag = SINUSOID
Historian and Access Server 1
2
3
SINUSOID

Figure 23: Example: Normal access

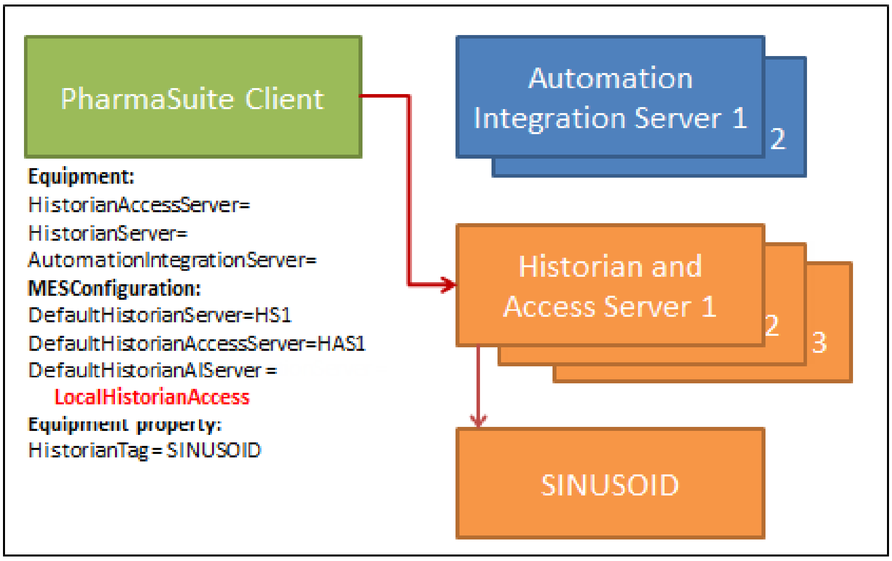

PharmaSuite Client
Automation Integration Server 1
2
Equipment:
HistorianAccessServer=
HistorianServer=
AutomationIntegrationServer=
MESConfiguration:
DefaultHistorianServer=HS1
DefaultHistorianAccessServer=HAS1
DefaultHistorianAIServer =
LocalHistorianAccess
Equipment property:
HistorianTag = SINUSOID
Historian and Access Server 1
2
3
SINUSOID

Figure 24: Example: Direct access for performance reasons

Reference Documents

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

No.	Document Title	Part Number
A1	FactoryTalk ProductionCentre 10.4 Supported Platforms Guide	PRDCTR-RM104A-EN-E
A2	FactoryTalk ProductionCentre Release 10.4 Database Installation Guide	PRDCTR IN104A EN E
A3	FactoryTalk ProductionCentre Plant Operations Release 10.4 Server Installation Guide - JBoss Advanced	PCJBAD IN104A EN E
A4	PharmaSuite Supported Platforms Guide	PSPG-RM010C-EN-E
A5	FactoryTalk ProductionCentre Administrator Release 10.4 User's Guide	PCADM-IN104A-EN-E
A6	PharmaSuite Technical Manual Configuration & Extension - Volume 4	PSCEV4-GR010C-EN-E
A7	RSLinx (http://www.rockwellautomation.com/rockwellsoftware/design/rslinx/)	N/A
A8	PharmaSuite Technical Manual Developing System Building Blocks	PSBB-PM010B-EN-E
A9	PharmaSuite Technical Manual Administration	PSAD-RM010C-EN-E
A10	PharmaSuite Implementation Guide PS Administration	PSAC-IN002C-EN-E
A11	PharmaSuite Technical Manual Configuration & Extension - Volume 1	PSCEV1-GR010C-EN-E
A12	PharmaSuite Technical Manual Installation - Building Blocks	PSBB-IN009C-EN-E
A13	FactoryTalk ProductionCentre Deployment Manager Administration Guide	DMGR AD042A EN E
A14	FactoryTalk ProductionCentre Modular Framework WebSDK User's Guide	WSDK UM042A EN E
A15	Enterprise Integration Hub - For PharmaSuite Installation and Configuration Guide	EIHPS-IN041A-EN-E
A16	Enterprise Integration Hub - PharmaSuite/FactoryTalk Warehouse Management Integration Guide	PSWM-IN041-EN-E
A17	Technical Manual Integration MES - WMS	VPWMS-GR005B-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 and other FactoryTalk installations.

No.	Document Title / Section
B1	Process Designer Online Help
B2	FactoryTalk Help This is the FactoryTalk Administration Console Online Help and the FactoryTalk Directory Server Location Utility Online Help.
B3	Live Data Test Client Help
B4	FactoryTalk ProductionCentre Support Home Page (http://rockwellsoftware.custhelp.com/app/home)
B5	FT Historian SE Installation and Configuration Guide, Rockwell Automation Publication HSE-IN025D-EN-E-July 2018

TIP

To access the "Process Designer Online Help", use the following syntax: *http://<MES-PS-HOST>:8080/PlantOperations/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	Production Execution User Documentation

TIP

To access the "Production Execution User Documentation", use the following syntax: *http://<MES-PS-HOST>:8080/PharmaSuite/documentationandhelp/index.htm*, 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	ActiveMQ, Tools - Java Service Wrapper (http://activemq.apache.org/java-service-wrapper.html)
D2	ActiveMQ - Features - Clustering - MasterSlave (http://activemq.apache.org/masterslave.html)
D3	OSIsoft (http://www.osisoft.com/) PI JDBC 2010 R3 Administrator Guide Configuring PI Server Security

Appendix A

Revision History

The following tables describe the history of this document.

Changes related to the document:

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

Changes related to "Introduction" (page 1):

Object	Description	Document
Introduction (page 1)	PharmaSuite version updated.	1.0

Changes related to "Installing PharmaSuite" (page 3):

Object	Description	Document
Creating Indexes on the Microsoft SQL Server Database System (page 13)	Updated point 9 of how to execute *Transact SQL* script on the FactoryTalk ProductionCentre production database: "Continue with enabling query store and activating forced query parameterization."	1.0
Enabling query store and activating forced query parameterization on the Microsoft SQL Server Database system (page 14)	Added a new section.	1.0
Configuring the ActiveMQ JMS Message Broker Failover Scenario (page 22)	Value of **MessageBrokerURL** configuration key extended with connection timeout setting.	1.1

Changes related to "Data Objects Provided Along with the Installation" (page 37):

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

Changes related to "Troubleshooting Installation Issues" (page 41):

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

Changes related to "Setting up the Live Data Infrastructure for PharmaSuite" (page 43):

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

Changes related to "Setting up the Historian Infrastructure for PharmaSuite" (page 63):

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

Index