

LISTEN.
THINK.
SOLVE.SM

INTEGRATED PRODUCTION & PERFORMANCE SUITE



Production Management

FactoryTalk[®] ProductionCentre



FACTORYTALK PRODUCTIONCENTRE ADMINISTRATOR
RELEASE 10.4
USER'S GUIDE

PUBLICATION PCADM-IN104A-EN-E

ALLEN-BRADLEY • ROCKWELL SOFTWARE

Rockwell
Automation

Contact Rockwell

Customer Support Telephone — 1.440.646.3434

Online Support — <http://support.rockwellautomation.com>

Copyright Notice

© 2017 Rockwell Automation Technologies, Inc. All rights reserved. Printed in USA.

This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Technologies, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies, Inc. is strictly prohibited. Please refer to the license agreement for details.

Trademark Notices

FactoryTalk, Foundation Server, Rockwell Automation, Rockwell Software, RSView, 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 View, FactoryTalk Administration Console, FactoryTalk Services Platform, FactoryTalk Live Data, RSAssetSecurity, FactoryTalk Security, FTPC Administrator, Integrate, Operational Data Store, ODS, Plant Operations, Process Designer, Live Transfer, Rockwell Software CPGSuite, Rockwell Software PharmaSuite, and Rockwell Software AutoSuite.

Other Trademarks

ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows ME, Windows NT, Windows 2000, Windows Server 2003, Windows 2008, and Windows 2012 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 anytime without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Table of Contents

Read Me First	7
Audience and Expectations	8
Organization	8
Other Information Sources	9
Product Installation Documentation Set	9
Related Documentation	9
Solutions and Technical Support	9
Hardware Requirements	9
Database User Login	10
 Chapter 1 FTPC Administrator Overview	 11
 Chapter 2 Live Transfer Overview	 13
Version Compatibility	14
Live Transfer Setup	15
Live Transfer Components	16
Site Extract Manager (SEM)	17
Table Extract Manager (TEM)	17
Table Loader Manager (TLM)	18
How Live Transfer Works	18
ActiveMQ Queue Throttling	20
PID Incrementation by Live Transfer	21
Extract Retry Processing	22
Database Log Updates	22

Chapter 3	Purge Overview	25
	Purge Limitations.....	27
	ActiveMQ Recommendations	27
	Tables Affected by Purge	28
	Instance Tables	28
	Application Tables	29
	Purge Processes	29
	Marker Process (Time-Based Purge)	30
	Marker Process (Status-Based Purge).....	30
	Reaper Process	32
	Aborting the Marker and Reaper Processes.....	32
	How Purge Works	33
	Time-Based Purge.....	36
	Time-Based Purge Criteria.....	36
	Status-Based Purge	39
	Status-Based Purge Criteria	39
	DCS Relationships	43
	Object Purge Criteria	44
	Batches	45
	ObjectStateHistory	47
	Process Orders	47
	Reports and Labels.....	48
	Stateless Runtime Objects.....	49
	Inventory Container History	50
	Tracked Runtime Objects.....	51
	Workflows.....	55
	Transactions	55
	Pallets	56
	Reverse the ATRows/Tracked Object Dependency	58
	Limitations	59
	Enabling the DeleteATRowsForClosedTrackedObjects Feature	
	59	
	SQL Throttle	63
Chapter 4	Installing and Running Live Transfer	65
	Installing Live Transfer.....	66
	Installing Live Transfer on Linux	69
	Configuring Live Transfer Installations.....	71
	Defining the BLOB Payload Size (Optional)	71
	Configuring Continuous Live Transfer (Optional)	72

	Increasing the Message Resend Interval (Optional)	72
	Registering and Starting Live Transfer Components.....	73
	Windows.....	73
	Linux	74
	Stopping and Unregistering Live Transfer Components	76
	Viewing Live Transfer Logs	77
Chapter 5	Installing and Running Purge	79
	Installing Purge	80
	Installing Purge on Linux.....	83
	Configuring Purge Installations	84
	Registering and Starting the Purge Service	85
	Windows.....	85
	Linux	86
	Stopping and Unregistering the Purge Service	88
	Viewing Purge Logs.....	89
Chapter 6	Installing FTPC Administrator	91
	Installation Prerequisites.....	92
	Installing and Configuring ActiveMQ	92
	Installing and Configuring Tomcat	92
	Undeploying the Previous Application	93
	Installing FTPC Administrator.....	93
	Installing on JBoss Standalone	93
	Installing on JBoss Advanced or WebSphere.....	93
	Configuring Data Management Installations	94
	Accessing FTPC Administrator	95
	Changing Passwords	97
Chapter 7	Working with Databases	99
	Production Database	100
	ODS Database	100
	Registering a Database.....	101
	Editing a Registration.....	105
	Deleting a Registration.....	105
	Initializing the Database	106
	Migrating a Database	107
	Migrating a Base ODS	108

Updating a Database	109
Reorganizing/Remapping a Database	110
Understanding Remapping.....	110
Understanding Reorganizing	111
Performing the Remapping/Reorganization	112

Chapter 8 Configuring Database Logging..... 115

Object Revision Logging.....	116
Understanding Primary and Audit Tables	117
How Primary and Audit Tables Work Together	118
Transaction Logging	118
History Logging.....	119
Configuring Database Logging.....	119
Configuring Auditing Options	121
Enabling/Disabling Auditing for Specific Tables	121
Disabling Runtime Activity Set Auditing	123
Configuring Application Logging.....	124
Viewing the Administrator Log	124

Chapter 9 Configuring System Features 127

Updating the Site Number	129
Setting Object Uniqueness.....	130
Enabling/Disabling Consumed Parts Consolidation.....	131
Enabling/Disabling Closing Consumed Lots.....	131
Setting the Client Inactivity Timeout.....	132
Configuring Access Control	132
Configuring the PID Increment Interval.....	133
Defining the Sublot Quantity Constraint	134
Enabling/Disabling Client-Side Distributed Event Broadcasting...	134
Enabling/Disabling Server-Side Distributed Event Broadcasting ..	136
Enabling/Disabling Keyed Object Changed Event Broadcasting ..	138
Defining JMS Connection Information	138
Configuring Messaging	139
Configuring Filtering.....	139
Enabling/Disabling SOAP Failover.....	140
Configuring Sublot Naming in Inventory Batches	140

Specifying Serial Numbers	141
Setting up Email Services	141
Modifying the User Login Requirements	142
Updating System Messages	143
Deleting Data	143
Deleting Runtime Data.....	144
Deleting Work Orders	144
 Chapter 10 Configuring Live Transfer	145
Creating Live Transfer Instances	146
Defining the Source Database.....	146
Defining the Destination Database	152
Creating Instances from Existing Configurations	153
Editing Live Transfer Configurations	154
Sending Live Transfer Configurations.....	154
Using the Comparison Tools.....	156
Using LTPIDCompare.....	156
Using LTSampleCompare	157
Starting and Stopping Live Transfer	158
Configuring Live Transfer Logging.....	159
Configuring ActiveMQ	160
Configuring a Separate ActiveMQ Cluster	160
Configuring ActiveMQ for a Firewall Environment.....	160
Configuring ActiveMQ Monitoring.....	161
Configuring Live Transfer with Linux.....	163
Setting the Maximum Memory	164
Configuring the Java Service Wrapper (Optional).....	164
Deployment Options and Recommendations.....	165
Extraction Recommendations	165
Loader Recommendations.....	166
ActiveMQ Recommendations.....	166
Deployment Scenarios	167
 Chapter 11 Configuring Purge	173
Creating a Purge Job	174
Configuring the Purge Service.....	180
Sending Purge Configurations	183
Starting and Stopping Purge Jobs	184
Running A Purge Job Manually	185

Viewing Purge Statuses	185
Updating Data Collection Maps (Status-Based Only).....	186

Chapter 12 Uninstalling and Upgrading Data Management..... 187

Uninstalling Live Transfer	188
Windows	188
Linux	189
Upgrading Live Transfer.....	190
Windows	190
Linux	190
Uninstalling Purge	190
Windows	191
Linux	192
Upgrading Purge	192
Additional Default Properties	192
Windows	193
Linux	193

Chapter 13 Troubleshooting 195

Understanding Live Transfer Messages.....	196
Recording Errors.....	197
Handling a Communication Loss	197
Encrypting Data	198

Index..... 199



Read Me First

In this chapter

- ❑ **Audience and Expectations** 8
- ❑ **Organization** 8
- ❑ **Other Information Sources** 9
 - Product Installation Documentation Set 9
 - Related Documentation 9
 - Solutions and Technical Support 9
- ❑ **Hardware Requirements** 9
- ❑ **Database User Login** 10

Audience and Expectations

This guide is intended for experienced professionals who understand their company's business needs as well as the technical terms and software dependencies described in this guide.

This guide covers installing the following:

- ❑ Installing FactoryTalk® ProductionCentre (called FTPC hereafter) Administrator.
- ❑ Configuring FTPC features using FTPC Administrator.
- ❑ Installing, configuring, and running Data Management components (Live Transfer and Purge) as part of FTPC.

This document assumes that the supporting network equipment and software, including the required databases, have been installed. This document does not provide installation instructions for related components, like database software or internet connections.

Organization

This book contains the following chapters:

- ❑ **Chapter 1, “FTPC Administrator Overview”** - Provides an overview of FTPC Administrator.
- ❑ **Chapter 2, “Live Transfer Overview”** - Provides an overview of Live Transfer.
- ❑ **Chapter 3, “Purge Overview”** - Provides an overview of Purge.
- ❑ **Chapter 4, “Installing and Running Live Transfer”** - Describes how to install and run Live Transfer.
- ❑ **Chapter 5, “Installing and Running Purge”** - Describes how to install and run Purge.
- ❑ **Chapter 6, “Installing FTPC Administrator”** - Describes how to install FTPC Administrator.
- ❑ **Chapter 7, “Working with Databases”** - Describes how to register, initialize, migrate, and reorganize databases using FTPC Administrator.
- ❑ **Chapter 8, “Configuring Database Logging”** - Describes how to configure database logging using FTPC Administrator.
- ❑ **Chapter 9, “Configuring System Features”** - Describes how to configure system features using FTPC Administrator.
- ❑ **Chapter 10, “Configuring Live Transfer”** - Describes how to configure Live Transfer.
- ❑ **Chapter 11, “Configuring Purge”** - Describes how to configure Purge.

- ❑ **Chapter 12, “Uninstalling and Upgrading Data Management”**- Describes how to uninstall and upgrade Live Transfer and Purge.
- ❑ **Chapter 13, “Troubleshooting”** - Provides troubleshooting tips.

Other Information Sources

In addition to this guide, you should review the following documentation.

Product Installation Documentation Set

This installation guide was designed to be used with the following other guides, available at the **Documentation** link on the FTPC software download web site (<http://download.rockwellsoftware.com>):

- ❑ *FactoryTalk ProductionCentre Supported Platforms Guide* for the latest supported software and FTPC product interoperability.
- ❑ *FactoryTalk ProductionCentre Database Installation Guide* for installing and configuring database software.
- ❑ *FactoryTalk ProductionCentre Plant Operations Server Installation Guide* for installing Plant Operations as well as installing and configuring Shop Operations Server.

Related Documentation

Please go to the appropriate websites for the required documentation needed to install any third-party products (for example, Microsoft SQL Server or Oracle).

Solutions and Technical Support

To contact Technical Support, call (440) 646-3434. Choose technical support for FTPC from the phone menu.

Hardware Requirements

The minimum hardware requirements for FTPC Administrator and Data Management are identical to the requirements for FTPC. The requirements may vary from site to site and should be reviewed to determine the appropriate configuration based upon the volume and use of the application for each site. Refer to the *FactoryTalk ProductionCentre Supported Platforms Guide* for the Plant Operations requirements.

Your databases must also have enough free space to meet the needs of your applications. The amount of free space you need depends on many factors,

including logging settings and the number of transactions being performed. Consult your Implementation Professional to estimate how much space you need.

Database User Login

The Data Management installation requires a database user login to connect to the Production and ODS databases and perform Live Transfer and Purge. The database user with the minimum required database privileges is described in the relevant section of the *FactoryTalk ProductionCentre Database Installation Guide*.

FTPC Administrator Overview

FTPC Administrator is a web-based user interface that can be accessed from a client machine. You can perform the following tasks using the FTPC Administrator:

- ☐ Initialize and migrate databases.
- ☐ Configure database logging.
- ☐ Configure and schedule Live Transfer and Purge.
- ☐ Modify security.
- ☐ Delete data.
- ☐ Set object uniqueness.
- ☐ Enable/disable consumed parts consolidation.
- ☐ Enable/disable closing consumed lots.
- ☐ Set the client inactivity timeout.
- ☐ Configure access control.
- ☐ Specify serial numbers.
- ☐ Define whether or not sublots can have negative quantities.
- ☐ Enable and disable subplot object revisioning.
- ☐ Set the Application Log retention period and maximum size.

There is a central “home” web server that is the starting point for an administrative session. From that central server, the administered components (such as Live Transfer and Purge services) are contacted. Each administered component can send and receive its configuration and respond to control requests, such as stop, start, restart, etc. The home server is responsible for presenting and managing the configurations and registrations of the administered components through FTPC Administrator.

Live Transfer Overview

In this chapter

- ❑ **Version Compatibility** 14
- ❑ **Live Transfer Setup** 15
- ❑ **Live Transfer Components** 16
 - Site Extract Manager (SEM) 17
 - Table Extract Manager (TEM) 17
 - Table Loader Manager (TLM) 18
- ❑ **How Live Transfer Works** 18
 - ActiveMQ Queue Throttling 20
 - PID Incrementation by Live Transfer 21
 - Extract Retry Processing 22
 - Database Log Updates 22

Live Transfer allows you to load data from multiple production or Operational Data Store (ODS) databases into a single target database (either an ODS or reporting database). The latency will depend on the number of source databases and the processing capacity of the servers and networks involved.

You can transfer data from a SQL Server database to an Oracle database in both Production-to-ODS and base ODS-to-hub ODS configurations. You cannot transfer data from an Oracle database to a SQL Server database. Therefore, once you transfer data from a SQL Server database to an Oracle database, all databases thereafter must be Oracle databases.

Live Transfer allows you to configure a specific list of database tables to be transferred if you do not want to transfer all the database tables in the source database. Use this feature if you are only interested in the data in certain tables. For example, you want to run reports against a specific set of tables. You can set up a Live Transfer that only transfers the data you are interested in to the target database, and then you can run your reports against the target database.

The database where these specific tables are transferred into is called a reporting database. Middle-tier and Purge operations are not supported for a reporting database because it is not considered to be an ODS. Only databases to which all of the tables are being transferred are considered to be an ODS.

IMPORTANT: If you are upgrading to 10.4 from a pre-8.2 release, before you can start using Live Transfer, you must configure and run the *populatePdXfrUpdatePid.bat* file to completion after you have migrated your ODS. When migrating an existing pre-8.2 ODS, the migration process creates a new column called *pd_xfr_update_pid* in any table that is loaded by Live Transfer. This column stores the original Production database's *xfr_update_pid* value. For the migrated ODS, this value is based on the information in the *XFR_LOAD_LOG* table. To initially populate the *pd_xfr_update_pid* column, you must configure and run the *populatePdXfrUpdatePid.bat* file after you have migrated your ODS. This file does not need to be run on newly-initialized ODSs.

See “Migrating a Base ODS” on page 108 for more information on the *populatePdXfrUpdatePid.bat* file and migrating your ODS.

Version Compatibility

Before configuring Live Transfer, note the following versioning limitations:

- ☐ The source database's Live Transfer schema version must be identical to the Site Extract Manager's expected Live Transfer schema version.
- ☐ The target database's Live Transfer schema version must be identical to the Table Loader Manager's expected Live Transfer schema version

For example, if the source database's Live Transfer schema version is 2, the Site Extract Manager's expected Live Transfer schema version must also be 2. Another example includes if the target database's Live Transfer schema version is 3, the Table Loader Manager's expected Live Transfer schema version must also be 3.

NOTE: When transferring into an FTPC10.4 target database, the source database must have an FTPC schema version of at least 92.05. The source database and target database are able to have the same schema version, but the source database cannot have a schema version higher than that of the target database.

For instance, you are able to transfer a Production database with a 9.2 version to a base ODS database with a 9.4 version. You can also transfer that same base ODS database with a 9.4 version to a hub ODS database with a 10.4 version. You cannot, however, transfer a Production database with a 10.4 version to a base ODS database with a 9.4 version.

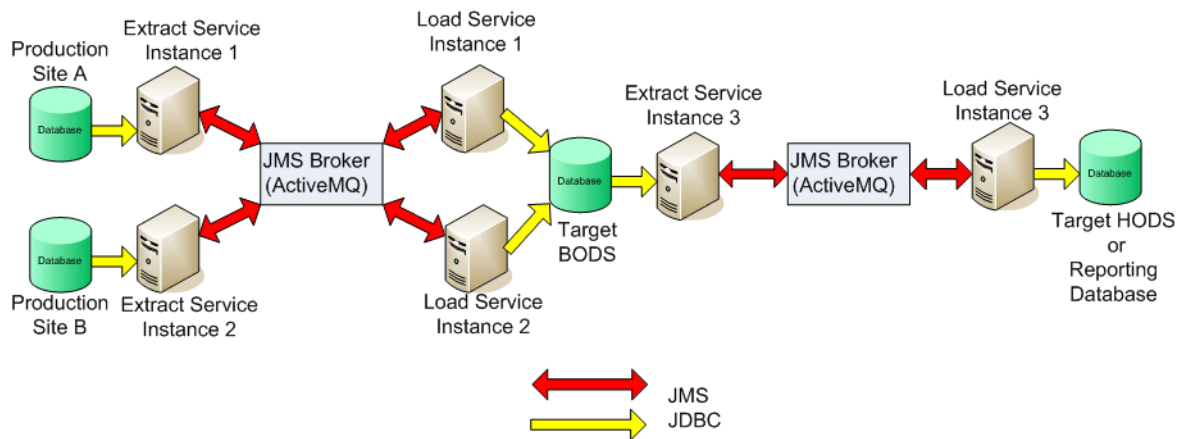
The schema versions for a database is recorded in the SCHEMA_VERSION table in the following rows:

- ❑ SCHEMA_VERSION: the Plant Operations major schema version.
 - ❑ SCHEMA_MINOR_VERSION: the Plant Operations minor schema version.
 - ❑ LIVE_TRANSFER_VERSION: Live Transfer schema version.
-

IMPORTANT: The Live Transfer schema version and the Production database schema version (major and minor) are two different version numbers that may, or may not be the same.

Live Transfer Setup

The following image shows how Live Transfer is set up:

Figure 2-1: Live Transfer Setup

- A single Extractor Service Instance can only extract from one source database. In the example figure, Extract Service Instance 1 only extracts from Production Site A, and Extract Service Instance 2 only extracts from Production Site B. Neither of these Extractor Service Instances can extract data from an additional database.
- The JMS Broker can range from a single instance of an Apache ActiveMQ (called ActiveMQ hereafter) broker to a network of brokers in a highly-available, load-balanced environment. Go to <http://activemq.apache.org/> for more information on ActiveMQ.
- Any given Load Service Instance can only load into a single target database, but multiple Load Service Instances can load into a single target database. In the example figure, Load Services Instances 1 and 2 both load into a single target database. They cannot load into any other database.
- A base ODS receiving data from Production databases can also transfer its data to a hub ODS or a reporting database.

Live Transfer Components

Live Transfer is made up of the following components:

- ☐ Live Transfer Extractor (LTE), which includes:
 - ▶ “Site Extract Manager (SEM)”
 - ▶ “Table Extract Manager (TEM)”
- ☐ “Table Loader Manager (TLM)”

Live Transfer uses JBossCache as a global memory space for keeping track of messages being sent to the loader as well as messages that indicate what has been successfully loaded by the loader. The TEM uses this information to avoid sending duplicate data to the loader, and the SEM uses this information to determine if a

table needs to be re-extracted as well as controlling when to send out the next extract requests.

JBossCache stores the Receipt object, which is used by the SEM to monitor an extract's progress. The SEM monitors the receipts and determines if all the extract requests have completed or if any one of them needs to be resent.

Site Extract Manager (SEM)

The Site Extract Manager (SEM) is the section of the Live Transfer Extractor component that determines the connection information for the source database and controls the extraction of data from the source database by dispatching messages to the extraction queue listing which tables have data that is ready to be transferred. The SEM also records the progress of the extraction and logs an extract log record in the source database when the transfer is complete.

To support this basic functionality, the SEM uses ActiveMQ and JBossCache to pass and share data with the other components in Live Transfer. The SEM is the main thread that keeps the communication between all the Live Transfer components open as well as recognizes when a new component has joined the network or when an extract or loader instance has been restarted.

Table Extract Manager (TEM)

The Table Extract Manager (TEM) is the section of the Live Transfer Extractor component that is responsible for extracting the data from the source database. It is made up of the following parts:

- ❑ **Manager:** This component performs the following tasks:
 - ▶ Communicates with the SEM upon startup to check compatibility issues.
 - ▶ Starts the Table Extractor instances.
 - ▶ Processes the load message receipts from the load service.
 - ▶ Updates JBossCache after the data is loaded successfully.
- ❑ **Table Extractor Instances:** This component performs the following tasks:
 - ▶ Queries for messages indicating that data is ready to be extracted.
 - ▶ Extracts the data from the source database.
 - ▶ Updates the global cache after data is extracted and sent to the Loader Queue.
 - ▶ Sends the extracted data to the Loader Queue where the loader is waiting to pick up the message from the queue.

Table Loader Manager (TLM)

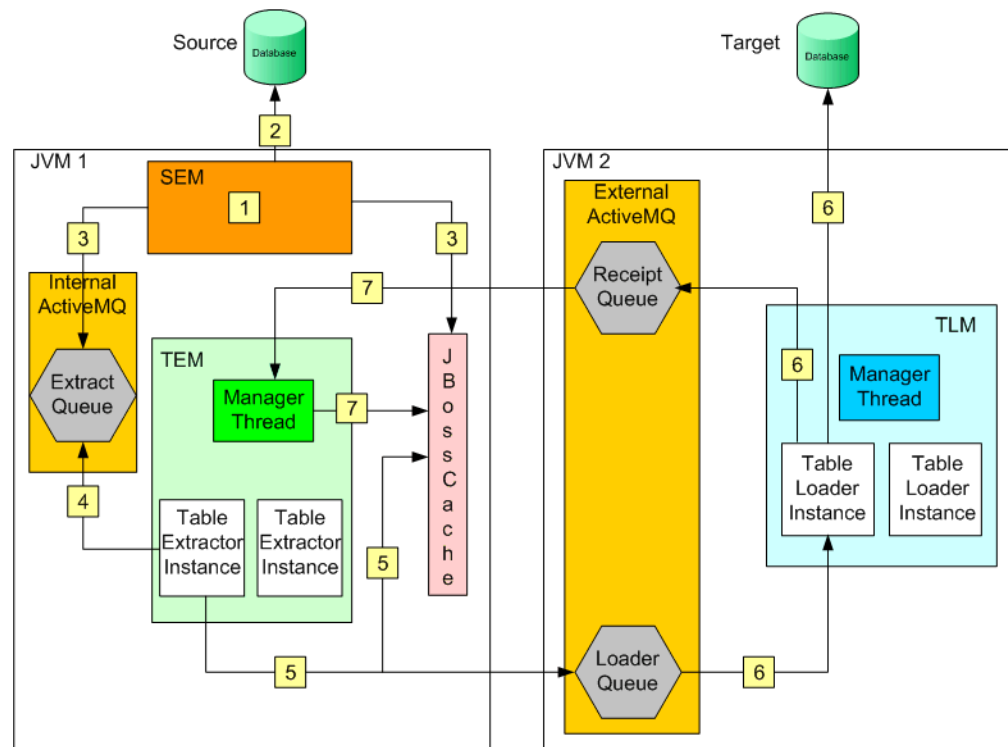
The Table Loader Manager (TLM) is the component responsible for loading the data into the target database. It is made up of the following parts:

- ❑ **Manager:** This component performs the following tasks:
 - ▶ Communicates with the SEM upon startup to check compatibility issues.
 - ▶ Starts the Table Loader instances.
 - ▶ Sends message receipts to the SEM when it has updated the logs after data has been loaded.
- ❑ **Table Loader Instances:** This component performs the following tasks:
 - ▶ Loads the extracted data into the target database.
 - ▶ Sends the load acknowledgements to the TEM manager when the data has been loaded.

How Live Transfer Works

The following diagram maps out how the components are used in Live Transfer. The two Java Virtual Machines (JVMs) can either be running on one physical machine or two different machines.

Figure 2-2: Live Transfer Overview



Step 1. The SEM, TEM, and TLM are started. The SEM communicates with the TLM and TEM to make sure that there are no compatibility issues. If no issues are found, then the TEM manager starts the Table Extractor instances and the TLM starts the Table Loader instances.

Step 2. The SEM checks the source database every 30 seconds to see if a new extract needs to be initiated by sending queries out that check to see if the process identifier (PID) has been incremented by the middle-tier. When applications create or update data, they store a PID with the data object. This allows the Live Transfer to handle the data in smaller sets and reduces extended periods of heavy load on the databases.

If the SEM detects that the PID has not changed since the last time it checked, then no extract will be initiated. The PID in a Production database is incremented by the middle-tier when a call is made to the middle-tier AND the PID increment period has elapsed (a default value of one hour) since the PID was last incremented. Change the increment interval value for the Production database using the *pidIncrementPeriod* item in the SITE_CONFIG database table. PIDs in ODSs are increased at the interval defined by the TLM's **PID Update Period** property and when Live Transfer is completed.

Step 3. If data needs to be extracted, the SEM sends a message to the Extract Queue indicating that an extract is required. One message is sent for each table that requires an extraction. For example, if the SEM finds that 10 tables require an extraction, 10 messages will be sent to the TEM manager.

When the SEM sends out the extract request, it also creates a Receipt object (one for each extract request) in the JBossCache.

Step 4. The Table Extractor instances query the extract queue for any messages from the SEM. Each message from the SEM in the extract queue is received by one Table Extractor instance. Therefore, each Table Extractor instance is responsible for extracting the information from one database table at a time because each message only contains the extract information for one table.

Step 5. If a message is received, the Table Extractor instance:

- a. extracts the data from the specified table. If one extract request covers a large amount of data, the Table Extractor instance may break the extract up into multiple extracts.
- b. updates the corresponding Receipt's **startTime** value in the JBossCache with the current time.
- c. sends a load message to the Loader Queue, specifying the data to be loaded. If one extract request was broken up into multiple extracts, a message for each extract is sent.
- d. updates the corresponding Receipt's **sentTime** value in the JBossCache with the current time.

Step 6. The Table Loader instance:

- a. receives the message from the Loader Queue.
- b. loads the data.
- c. posts the acknowledgment onto a Receipt Queue.

Step 7. The TEM's manager thread:

- a. receives the acknowledgment message from the Receipt Queue.
- b. updates the corresponding Receipt's **loadedTime** value in the JBossCache with the current time.

ActiveMQ Queue Throttling

To keep from overflowing the ActiveMQ queue with extract messages, the **maxUnacknowledgedMessages** property in the *SiteExtractManager.properties* file determines the minimum number of unacknowledged extract messages that can be in the ActiveMQ queue per Table Extractor before another extract message is sent. The default is 4 with a minimum value of 1. This means that by default, once the maximum number of unacknowledged messages has been reached (i.e., 4), the Table Extractor will wait and not send another extract message to the TLM until the number of unacknowledged messages in the queue has dropped to zero. Messages are processed out of the queue in a first-in-first-out (FIFO) order. Messages will not be resent until the time defined by the **reSendTimeout** property has elapsed.

Live Transfer is affected by the **maxUnacknowledgedMessages** property as follows:

- ❑ Because the flow to the loader is throttled by using the **maxUnacknowledgedMessages** property to avoid overflowing the ActiveMQ queue, the TLM has time to finish loading each message.
- ❑ The SEM monitors the current extract for completion. A new extract request will not be sent out unless both the current extract is complete AND the **extractPeriod** has expired. For example:
 - ▶ If the **extractPeriod** is 30 minutes and the current extract ran to completion from 2:00 PM to 2:20 PM, no extract request will be sent because the **extractPeriod** has not elapsed since the last extract request (Current Time – Last Extract Request Time = 20 minutes).
 - ▶ If the **extractPeriod** is 30 minutes and the current extract has not completed, then no extract request will be sent because the current extract has not completed.

TIP: No configuration changes are needed to enable this throttling behavior. Live Transfer will perform these functions by default out-of-box.

PID Incrementation by Live Transfer

As stated in [step 2](#) of the previous section, the Live Transfer PID is incremented by the middle-tier when a call is made to the middle-tier AND the PID increment period has elapsed since the PID was last incremented. However, if you are running Live Transfer during a normal 8:00 - 5:00 work day, the PID may be incremented by Live Transfer itself if a call to the middle-tier is not being made at the end of the day.

For example:

1. The last extract was at 4:45, and the PID increment period is 15 minutes.
2. The work day ends at 5:00. In the last 15 minutes of the day, no call has been made to the middle-tier.
3. The next PID increment is expected to occur at 5:00. However, because no call to the middle-tier has been made, the middle-tier does not increment the PID.

In this instance, Live Transfer will increment the PID and trigger the final extract of the day if the following properties are true:

- ❑ The middle-tier has not performed any transactions after the last extract.
- ❑ One minute has elapsed. This is the amount of time that needs to elapse before Live Transfer initiates an extract (30 seconds) times two.

Using the previous example, Live Transfer increments the PID, which will initiate an extract at 5:00 (which is the time after waiting an interval of one minute), because:

- ☐ The middle-tier has not performed any transactions after the last extract.
- ☐ One minute has elapsed since the last extract.
- ☐ Live Transfer has not incremented the PID since midnight.

Live Transfer will only increment the PID and start an extraction **once** within a 24-hour period starting at midnight. After that, the next extraction will not occur until the normal requirements of the PID increment period has elapsed AND a call to the middle-tier has been made. Therefore, after Live Transfer has initiated an extract at 5:00 (the time after waiting an interval of one minute), another extract will not be initiated until another call is made to the middle-tier (for example, a user logging in for the first time in the morning).

Extract Retry Processing

The SEM will re-issue a message to the extract queue in the following situations:

- ☐ If the SEM finds a message indicating an extraction was performed but does not find a corresponding message indicating that the load was successful, then it will re-issue a message to the extract queue for the table that was not successfully loaded, and the transfer process starts again for that table.
- ☐ If the SEM finds that an extract was started but never completed and, therefore, no message was sent to the loader, then the SEM will re-issue the extract request after a certain amount of time has elapsed as defined by the **Resend Timeout** property (four minutes by default).

Database Log Updates

While Live Transfer is running, the SEM queries the JBossCache to make sure that the extracts and loads are successful. The database logs are updated differently depending on the transfer mode:

- ☐ **All tables are being transferred (ALL mode):** When the SEM confirms that all of the data for all of the tables has been successfully loaded for a given PID, the following occurs:
 - a. The SEM updates the XFR_EXTRACT_LOG in the source database with the transfer progress information.
 - b. The SEM sends a message to the TLM manager indicating that records need to be inserted into the XFR_LOAD_LOG in the target database to record the transfer progress information.
 - c. The TLM manager updates the target database's XFR_LOAD_LOG accordingly and sends a message back to the SEM.
 - d. The SEM marks the transfer as complete in the XFR_EXTRACT_LOG.

- ❑ **Specific tables are being transferred (LIST mode):** When the SEM confirms that all of the data for the specified tables has been successfully loaded for a given PID, the SEM updates the LT_EXTRACT_TABLE_LOG in the source database with the transfer progress information.

NOTE: If you are migrating to 10.4 and had previously used Historical Transfer, please note that when populating the XFR_EXTRACT_LOG table, the available statuses differ between Historical Transfer and Live Transfer. Historical Transfer has three status values available: extractInProgress, extractCompleted, and loadCompleted. Live Transfer has two status values available: extractInProgress, and loadCompleted.

Purge Overview

In this chapter

- ❑ **Purge Limitations** 27
- ❑ **ActiveMQ Recommendations** 27
- ❑ **Tables Affected by Purge** 28
 - Instance Tables 28
 - Application Tables 29
- ❑ **Purge Processes** 29
 - Marker Process (Time-Based Purge) 30
 - Marker Process (Status-Based Purge) 30
 - Reaper Process 32
 - Aborting the Marker and Reaper Processes 32
- ❑ **How Purge Works** 33
- ❑ **Time-Based Purge** 36
 - Time-Based Purge Criteria 36
- ❑ **Status-Based Purge** 39
 - Status-Based Purge Criteria 39
 - DCS Relationships 43
- ❑ **Object Purge Criteria** 44
 - Batches 45
 - ObjectStateHistory 47
 - Process Orders 47
 - Reports and Labels 48
 - Stateless Runtime Objects 49
 - Inventory Container History 50
 - Tracked Runtime Objects 51
 - Transactions 55
 - Workflows 55
 - Pallets 56

- ❑ **Reverse the ATRows/Tracked Object Dependency** 58
 - Limitations 59
 - Enabling the DeleteATRowsForClosedTrackedObjects Feature 59
- ❑ **SQL Throttle** 63

Administrators can purge data from both the Production database and ODS using one of the following types of Purge:

- ❑ “Time-Based Purge” - records are deleted based solely on retention periods.
- ❑ “Status-Based Purge” - records are deleted based on object type and the specific criteria for each object type.

NOTE: Purge 10.4 is only supported with databases that have an FTPC 10.4 schema version. You cannot run Purge 10.4 with databases that have a different version of FTPC installed.

To set up a Purge job, you must configure the following

1. Select the databases that will be purged.
2. Define the Purge job’s properties.
3. Define the Purge job’s schedule.

See [Chapter 11, “Configuring Purge”](#) for more details.

Purge Limitations

Purge has the following limitations:

- ❑ A status-based Purge job can be configured and run for a database that has run a time-based Purge job **if** the time-based Purge job has been deleted.
- ❑ A time-based Purge job **cannot** be configured and run on a database that has already run a status-based Purge job.
- ❑ Multiple Purge jobs can be scheduled to run at the same time, but each task should be running against a different database.

ActiveMQ Recommendations

The Purge installer provides a service for running ActiveMQ. If you already have ActiveMQ installed, please use the Rockwell-provided ActiveMQ package for FTPC purposes.

We recommend that you configure a single ActiveMQ instance that will be used by all of the FTPC components, such as Live Transfer and Purge. In order to avoid this single instance becoming a single point of failure, the instance should be in a failover-configured and load-balanced ActiveMQ cluster.

TIP: Follow the instructions at <http://activemq.apache.org/shared-file-system-master-slave.html> to set up your shared file system master slave environment for ActiveMQ. Please use the configuration where the persistence adapter is defined by the KahaDB directory.

You do not need to configure any queues in ActiveMQ. If a client attempts to listen or send to a queue that does not exist, ActiveMQ will create the queue automatically. The Purge Request Queue is named by the *purge-instance-name* value defined in the *jms-local.config* file.

Tables Affected by Purge

When a Purge job is run, only the rows that contain runtime objects in the database tables and their corresponding audit tables are affected. In other words, one statement from the marker process can result in two delete statements (one for the primary table and one for the audit table) by the reaper process (See “[Purge Processes](#)” on page 29 for more information on the marker and reaper processes.). For example, a marker statement that calls for a unit to be deleted from the UNIT table will result in the reaper issuing a delete statement to the UNIT_A table as well as the UNIT table. Please note that while a single row is deleted from the UNIT table, multiple rows may be deleted from the UNIT_A table based on the number of audit records that exist for the unit being deleted.

The rows that correspond to buildtime objects are not purged. For example, even though the BOM table is affected by a Purge job, the BOM objects created in Process Designer are not purged. Only the runtime BOMs associated with work orders are purged. Records for that work order BOM in the UDA_BillofMaterials table are also purged.

NOTE: The **PurgeFetchPurgeRecords** parameter (defined in the DefaultServerJobConf.xml file) controls how many records are retrieved from the marker’s statement file in a single fetch, but the number of delete statements logged by the reaper in the DsDataManagementServerLog.html file is the number of delete statements that were produced from the fetch from the marker’s statement file. For example, if each row in the statement file results in two delete statements (one for the primary table and one for the audit table), then the number of delete statements will be twice as large as the number of statements in the statement file.

Instance Tables

Even though instance tables (e.g., AT_*, DC_*, STA_*, and UDA_* tables) in general are affected by Purge, not every instance table will be affected. If you are running a status-based Purge job, only those tables whose parent object class is

purgeable will be affected (e.g., a Part object is not purgeable so UDA_Part will not be purged). If you are running a time-based Purge job, the instance tables will be purged regardless of whether the parent object can be purged (e.g., a Work Order object is not purgeable, but the UDA_Order table can still be purged.).

Application Tables

Whether or not the runtime data for an application table is purged depends on the runtime data's management type, purge status, and the parent application table's purge status.

- ❑ If the runtime data for an application table is marked as non-static data (`dataManagementType = purgeable`) AND the ATRow's purge status is set to true [`ATRow.setPurgeStatus(true)`], then the data will be deleted during scheduled Purge jobs if the retention period has been satisfied.
- ❑ If the runtime data for an application table is marked as static data (`dataManagementType = static` or `exportable`) OR the ATRow's purge status is set to false [`ATRow.setPurgeStatus(false)`], then the data will not be deleted during scheduled Purge jobs.
- ❑ When all the parent and dependent application table rows are configured as purgeable [`dataManagementType = purgeable` and `ATRow.setPurgeStatus(true)`], then when a dependent application table row is purged depends on the purge status of the parent application table row regardless of the dependent application table row's purge status. The dependent application table row cannot be purged unless its parent application table row is purgeable.

You can configure whether or not the runtime data is purgeable when you configure the application table definition's **dataManagementType** property and by setting the ATRow's purge status [`ATRow.setPurgeStatus(boolean)`].

Purge Processes

Purge consists of two processes, the marker and the reaper. In a Production database, these processes will not run if they detect that other marker or reaper processes are running. Multiple processes can run in parallel in the ODS as long as they are running against different site numbers. Purge processes do not yield to any Live Transfer processes because Purge and Live Transfer do not interfere with each other by design.

If the marker and reaper processes will be running on different systems, note the following versioning limitations:

- ❑ The build numbers of the systems running the processes must match.
- ❑ The production and ODS databases must have the same Plant Operations database schema version.

The schema versions for a database is recorded in the `SCHEMA_VERSION` table in the following rows:

- ❑ `SCHEMA_VERSION`: the Plant Operations major schema version.
- ❑ `SCHEMA_MINOR_VERSION`: the Plant Operations minor schema version.

Marker Process (Time-Based Purge)

The marker process in a time-based Purge job identifies rows or objects to purge by using the retention period and the `xfr_update_pid` value. The rows are marked by inserting Purge records into the `PURGE_RECORD` table. A Purge record stores the object key and the complete `DELETE` statement that can be used to delete the marked row later by the reaper. When the marker marks a row, the row's corresponding rows in the audit table are also marked.

Marker Process (Status-Based Purge)

The marker process in a status-based Purge job identifies data to be purged. There are two types of markers that are used in the status-based Purge process. Which one your status-based Purge job uses depends on how you configured your job.

Production Marker

This marker consists of four subprocesses that run from the Production datasource:

- ❑ One marks tracked runtime objects.
- ❑ One marks stateless runtime objects.
- ❑ One marks transactions.
- ❑ One marks workflows.

To select the objects that will be purged, the marker uses the retention period and the `xfr_update_pid` value of the rows to mark transactions, stateless runtime object, and objects other than the tracked objects. For tracked objects, the marker uses the tracked status, the Purge window determined by the retention period, and the Purge rules listed in “[Tracked Runtime Objects](#)” on page 51 to select Purge candidates.

As in time-based Purge, the marker only issues `SELECT` statements to find rows. Rows are marked by inserting Purge records into the `PURGE_RECORD` table. A Purge record stores the object key and the complete `DELETE` statement that can be used to delete the marked row sometime later by the reaper.

ODS Marker

The ODS marker runs on the ODS, which reduces the processing load on the Production database server. The marking results are stored in a file saved in the locations defined by the **Marker File Directory** and **Reaper File Directory** properties. See “[Configuring the Purge Service](#)” on page 180 for details.

NOTE: The ODS marker can only be used to mark records in a Production database or to mark records in the same ODS that is running the marker process. Do not run the ODS marker on one ODS to mark records in another ODS. For example, do not run the marker on a base ODS to mark records in a hub ODS.

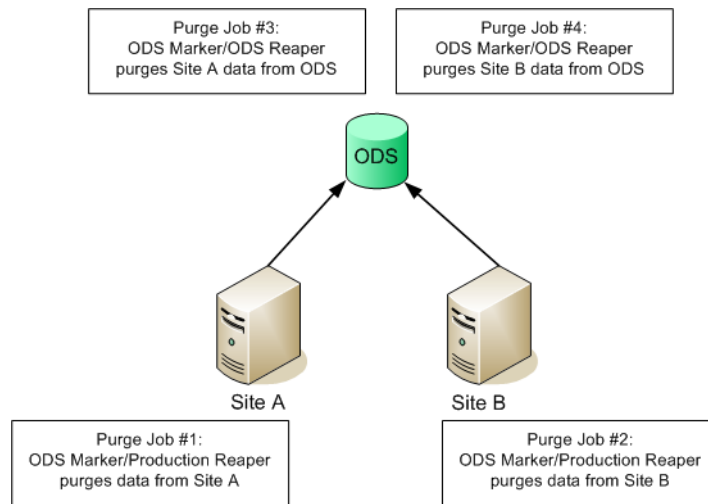
The ODS marker consists of the following subprocesses:

- ☐ One marks batches.
- ☐ One marks reports and labels.
- ☐ One marks tracked runtime objects.
- ☐ One marks ObjectStateHistories.
- ☐ One marks stateful runtime objects excluding labels and reports.
- ☐ One marks stateless runtime objects.
- ☐ One marks transactions.
- ☐ One marks workflows.

The ODS Marker records the marking results in a file that is transmitted to the reaper.

IMPORTANT: The ODS marker implements all of the functionality of the Production marker, except that it does not implement a Purge window to determine purgeability. This means that stateful runtime objects in earlier PIDs that were not Purge candidates are revisited and are considered for deletion in subsequent Purge jobs. For stateful runtime objects, the marker uses the data retention period as well as the specific system-defined semantic properties defined for that object. If you want the ODS marker functionality, you must configure your status-based Purge job accordingly. To do so, see [Chapter 11, “Configuring Purge”](#).

When using the ODS marker, the Purge jobs are site-specific. For example, you have two Production databases being purged using the ODS marker. To purge the data, you must create a Purge job for each Production database. In [Figure 3-1](#), both Site A and Site B are Production databases whose markers are run on the same ODS. To mark the records for both sites, one Purge job is configured for Site A and another Purge job is configured for Site B.

Figure 3-1: Purging Multiple Databases with the ODS Marker

If Live Transfer is used to transfer data from two different sites (Site A and Site B) into a single ODS, purging the ODS will also require two separate Purge jobs, one job to purge data from Site A from the ODS and one job to purge data from Site B from the ODS. These jobs need to be executed from a single purge process.

Reaper Process

The reaper deletes the marked rows from the database tables. The reaper reads the Purge records from either the `PURGE_RECORD` table (reaping using the Production marker), the file stored in the location defined by the *PurgeMarkerPath* property (local reaping using the ODS marker), or the file stored in the location defined by the **Reaper File Directory** property (remote reaping using the ODS marker) and issues the stored DELETE statements to delete the marked rows. After the purge, the Purge records are deleted from the table. The reaper terminates when there are no more Purge records.

Because the reaper issues DELETE statements in a tight control loop, its processing must be governed to control the load it puts on the application and database servers. The load on the application server can be controlled by setting the reaper wait time (the time it must wait after processing a Purge record before processing another). The load on the database server can be controlled by the SQL Throttle. See “[SQL Throttle](#)” on page 63 for more information.

Aborting the Marker and Reaper Processes

The marker process will be aborted if any of the following situations are found:

- ☐ In the Production database:
 - ▶ The retention period cutoff date occurs before the database was initialized.

- ▶ Purge has never run and the first interval start ID (starting at 2) includes data that still falls within the retention period.
 - ▶ Purge has been run, but the next interval start ID includes data that still falls within the retention period.
 - ▶ Live Transfer has been run and the next interval start ID includes data that has not been transferred.
- ❑ In the ODS database:
- ▶ No data was loaded into this ODS before the cutoff date dictated by the retention period.
 - ▶ Purge has never run and the first interval start ID includes data that still falls within the retention period.
 - ▶ Purge has been run, but the next interval start ID includes data that still falls within the retention period.
 - ▶ This ODS transfers data to another HODS and the next interval start ID includes data that has not been transferred.

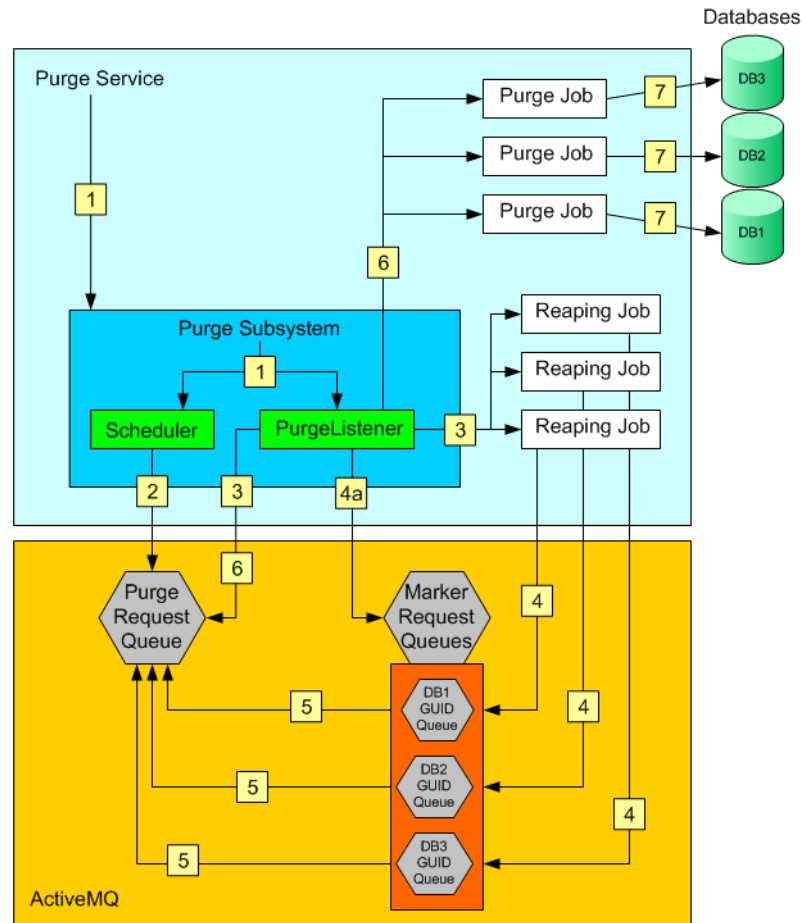
Whenever one of the above conditions occurs, the Purge log entry will have a *markerAborted* status. However, because this status does not necessarily indicate an error condition, the marker process will only be aborted for the specified object type. This would not prevent markers for other object types from running and, more importantly, will not prevent the reaper from running.

If either the marker or reaper process fail, you will get a status of *purgeFailed*. See “Viewing Purge Logs” on page 89 for more details.

How Purge Works

The following diagram maps out how Purge works.

Figure 3-2: Purge Overview



- Step 1.** When the Purge Service is started, it creates an instance of the Purge Subsystem, which starts the Scheduler and PurgeListener.
- Step 2.** The Scheduler examines the schedules configured in FTPC Administrator. When a job needs to be run, it sends a message to the Purge Request Queue. ActiveMQ is used to pass and share data between the Purge components.
- Step 3.** The PurgeListener processes the Purge request in the Purge Request Queue.
- ▶ If the Purge job is either a time-based Purge job or a status-based Purge job where both the reaper and marker processes are run on the Production database, then Purge job instances are created using the configuration defined in FTPC Administrator that purge the defined databases. Continue to [step 7](#).
 - ▶ If the Purge job is a status-based Purge job where the marker runs on the ODS, then the PurgeListener creates reaping Purge jobs. The reaper Purge jobs contain the following information:
 - ❖ The GUID of the database to be reaped.

- ❖ The GUID of the ODS database where the marker is run.

Continue to the next step.

Step 4. The reaping Purge jobs send requests to the marker request queue that corresponds to the GUID of the database that the job is supposed to reap. The marker request queue for that database may be on the local Purge Service or on a remote Purge Service.

The PurgeListener also monitors the Marker Request Queues because the names of the Marker Request Queues are based on the site GUID of the configured databases as provided by the PurgeListener. These queues are created by the PurgeListener (Step 4a).

Step 5. Marking information is sent back to the Purge Request Queue.

Step 6. The PurgeListener processes the marking information and creates the Purge job instances using the configuration defined in FTPC Administrator. These instances will mark the records to be purged in the databases.

Step 7. Once the records are marked, the Purge jobs delete the marked records in their corresponding databases.

Please note the following behavior when running Purge:

- ☐ A status-based Purge job can be configured and run for a database that has run a time-based Purge job if the time-based Purge job has been deleted.
- ☐ If you run status-based Purge on a database where the last Purge job used time-based criteria, the status-based Purge job will begin purging data where the time-based Purge job left off. For example, if the last record purged using time-based criteria had an *xfr_update_pid* of 10, when you run status-based Purge next, the records deleted will have *xfr_update_pid* values that are both greater than 10 and older than the retention period. If you want the status-based Purge run to delete records containing all PID values beginning with 2, you must create a status-based Purge job where the marker runs in the ODS datasource. Please refer to [Chapter 11, “Configuring Purge”](#) for more information.
- ☐ A time-based Purge job cannot be configured and run on a database that has already run a status-based Purge job
- ☐ If you run a status-based Purge job on an ODS where no Production-to-ODS Purge job has run, data that still exists in the Production database could be deleted from the ODS. For this reason, we recommend that you perform a Production-to-ODS Purge job at least once before you begin purging data from the ODS.
- ☐ A database can only have one configured Purge job (either time-based or status-based) running at a time. Multiple Purge jobs can be scheduled to run at the same time, but the second job’s process will only begin when the first job’s process has finished.

Time-Based Purge

This Purge process allows the administrator to select when to delete a record based solely on three retention periods configured for each type of data listed below:

- ☐ runtime objects
- ☐ test and data collection objects
- ☐ transaction

The retention periods are the number of days that different types of data are retained in the Production and ODS databases before being deleted. The retention periods for your Production database do not have to be the same as the retention periods for your ODS database. Data older than these settings is deleted (purged) during the Purge operation. The minimum retention period you can set for any data is one day.

NOTE: Time-based Purge does not support the process model. Process orders can only be marked for purging by using the ODS marker.

If you are running Live Transfers, PIDs will not be purged until they have been successfully loaded into an ODS, regardless of the configured data retention periods.

If you want to keep all data related to an object (its test data as well as its runtime data, for example), make sure that the data retention periods for all types of data regarding that object are set to purge at the same time because it is possible for data regarding an object to be purged while the object is not.

TIP: See Knowledgebase Article 2953 for more information on time-based Purge that will help you determine if it is the correct choice for your environment. Contact Rockwell Technical Support for access to the FTPC Knowledgebase.

Time-Based Purge Criteria

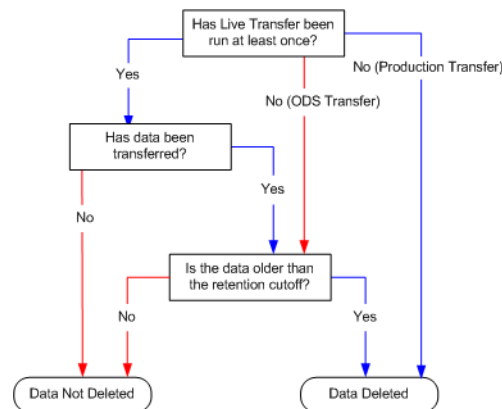
The time-based Purge process deletes data using the following criteria:

1. Has Live Transfer run at least once?
 - ▶ If not and you are purging from a Production database, then data is deleted.
 - ▶ If not and you are purging from an ODS, then the data retention period is checked and if the data is older than the retention period, then it is deleted.

If Live Transfer has run at least once, the Purge process will never purge any data that has not been transferred.

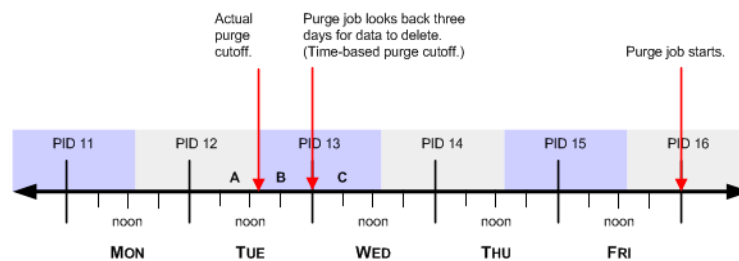
2. If the data has been transferred to an ODS, is the data older than the data retention cutoff? If it is, then the data will be deleted. Otherwise, it will not be deleted.

Figure 3-3: Time-Based Purge Criteria Flowchart



This Purge process uses the PID value associated with objects to determine if the object is older than the current data retention cutoff. If the cutoff time defined by the configured data retention period falls between PID increments, then the Purge process will not delete objects with that PID and will use the prior PID as a cutoff instead.

Figure 3-4: Purge and PID Example



In [Figure 3-4](#), the system starts a Purge job on Saturday at midnight. The purge is configured with a three-day object retention period. Points A, B, and C represent the last time data objects were modified. The PID is incremented at 1:35 PM. As data objects are saved, they are associated with the current PID.

The Purge job starts and considers a point in time three days earlier, Wednesday at midnight. Data objects A and B were modified before this time and object C was modified after. Since the cutoff time occurs between PID increments, it falls in the range of data objects associated with PID 13. The Purge process will not delete a subset of objects with the same PID and, instead, uses the previous PID, PID 12, as the cutoff for purging data. The Purge process deletes data object A and does not delete objects B and C.

By using PID intervals, the Purge process can run without taxing the database system or deadlocking with the ongoing operation of the active operational system

because information can be deleted in smaller batches. This allows the practical running of the Purge program in a 24x7 operation.

NOTE: Data objects are purged based only on this criteria. For example, a unit object will be deleted from the database if it meets this criteria regardless of whether the unit is closed or in an open box. Furthermore, a UnitHistory object that has not been modified for a period of time that is longer than the runtime business object retention period will be deleted, even if the unit itself is still being processed and has been modified recently.

The same rules used in purging the Production database apply to purging records from the ODS except the two databases may have different retention intervals. If Live Transfer has been run at least once between a BODS and an HODS, then untransferred data cannot be deleted from a BODS.

Status-Based Purge

The status-based Purge allows the administrator to select when to delete a record based on its object type. Each object type has its own purge criteria as defined in the following table:

Table 3-1 Status-Based Purge Criteria by Object

	object status	exit time	last modified time	retention period
Reports and Labels			X	X
ObjectStateHistory Objects		X	X	X
Process Orders	X		X	X
Stateless Runtime Objects ^a			X	X
Workflows	X			X
Transactions				X
Tracked Runtime Objects	X		X	X
Batches	X		X	X
Inventory Container History ^b			X	X
Pallet	X		X	X
^a In the case of ATRows, if the ATRow references a unit, lot, order, or order item, you can reverse the dependency and make the ATRow purgeable if the referenced object is purgeable regardless of the ATRow's purge status and retention period configuration. See Reverse the ATRows/Tracked Object Dependency for more details. ^b The current inventory container history record is never purged regardless of the retention period.				

NOTE: Within each row, the checked items are AND, not OR. For example, reports and labels are purged based on last modified time AND retention period, not last modified time OR retention period.

See “Object Purge Criteria” on page 44 for more details.

NOTE: If you are running Live Transfers, PIDs will not be purged until they have been successfully loaded into an ODS, regardless of the configured data retention periods.

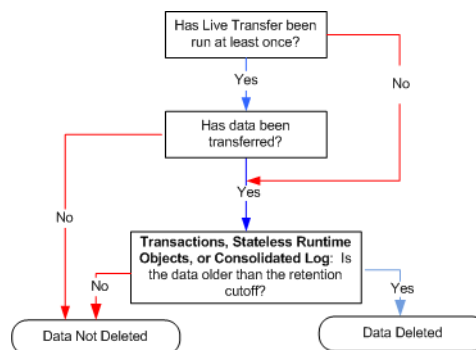
Status-Based Purge Criteria

The status-based Purge process deletes data using the following criteria:

1. Has Live Transfer been run at least once? If not, then the purge requirements for each object type are checked (see [step 3](#)).

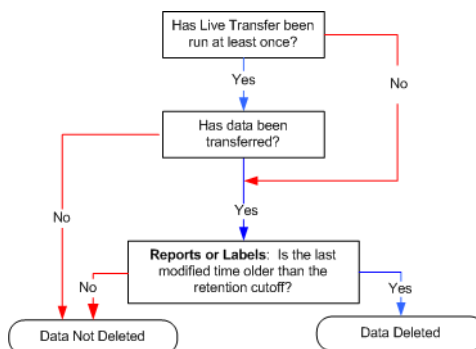
2. If Live Transfer has been run at least once, has the data object been transferred to the ODS? If not, then the data object will not be deleted. If Live Transfer has run at least once, the Purge process will never purge any data that has not been transferred.
3. The purge requirements vary according to object type:
 - If you are trying to purge a transaction or stateless runtime object (other than labels and reports), or a consolidated log record, is the data older than the data retention cutoff? If so, then it will be purged.

Figure 3-5: Purging Transactions, Stateless Runtime Objects, and Consolidated Log

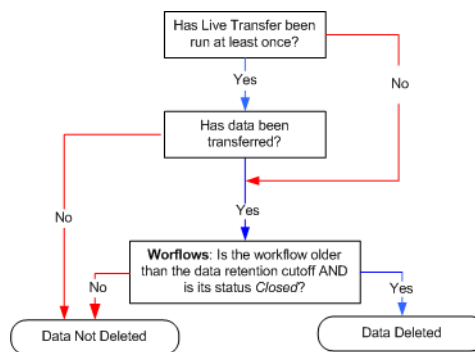


- If the object is a report or label, is the object's last modified time older than the data retention period? If so, it will be purged.

Figure 3-6: Purging Reports and Labels

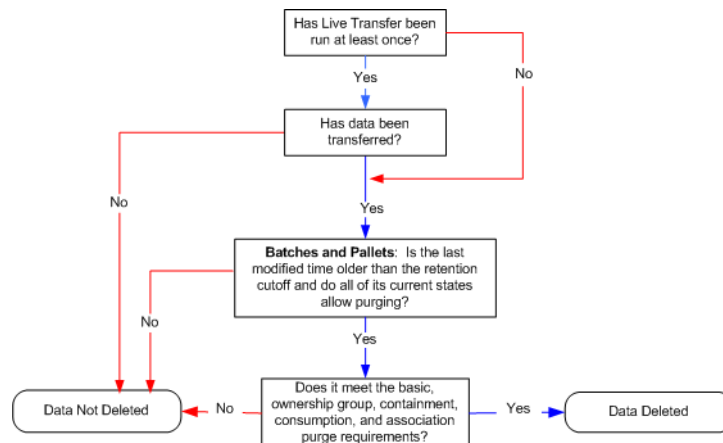


- If the object is a workflow, is the workflow older than the data retention cutoff and is its status *Closed*? If so, it will be purged.

Figure 3-7: Purging Workflows

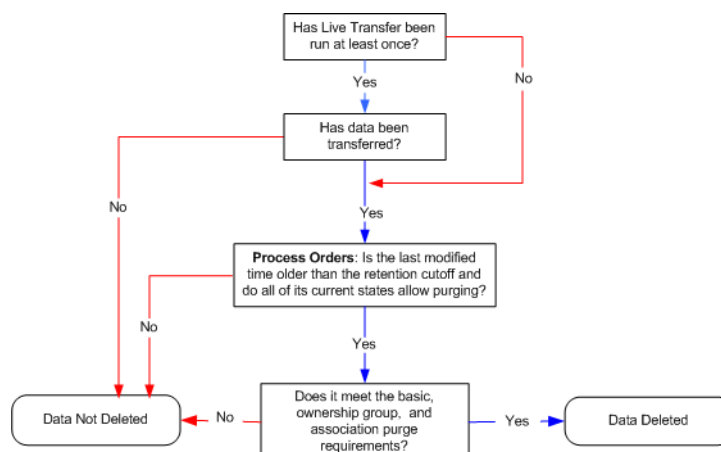
- If the object is a batch or a pallet:
 - ❖ Is the object's last modified time older than the data retention period and do all of its current states allow purging?
 - ❖ Does it meet the basic, ownership group, containment, consumption, and association purge requirements?

If the answer to both questions is yes, then the object is purged. All of these criteria must be met for the object to be purged.

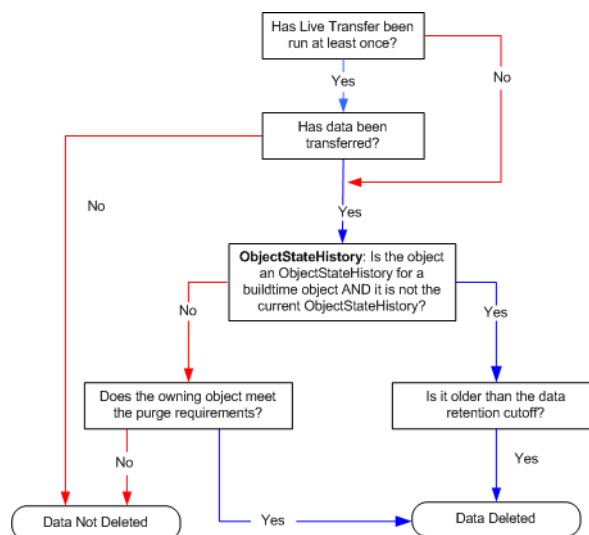
Figure 3-8: Purging Batches and Pallets

- If the object is a process order:
 - ❖ Is the object's last modified time older than the data retention period and do all of its current states meet the purge requirements?
 - ❖ Does it meet the basic, ownership group, and association requirements?

If the answer to both questions is yes, then the object is purged. All of these criteria must be met for the object to be purged.

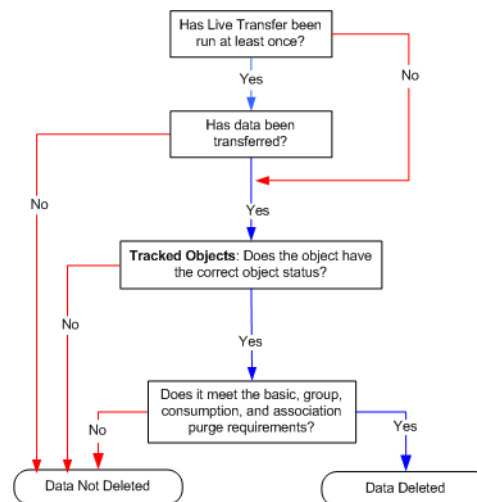
Figure 3-9: Purging Process Orders

- ▶ If the object is an ObjectStateHistory object for a runtime object, does the owning object meet the purge requirements? If so, it will be purged.
- ▶ If the object is an ObjectStateHistory for a buildtime object and it is not the current ObjectStateHistory, is it older than the data retention cutoff? If so, it will be purged.

Figure 3-10: Purging ObjectStateHistory Objects

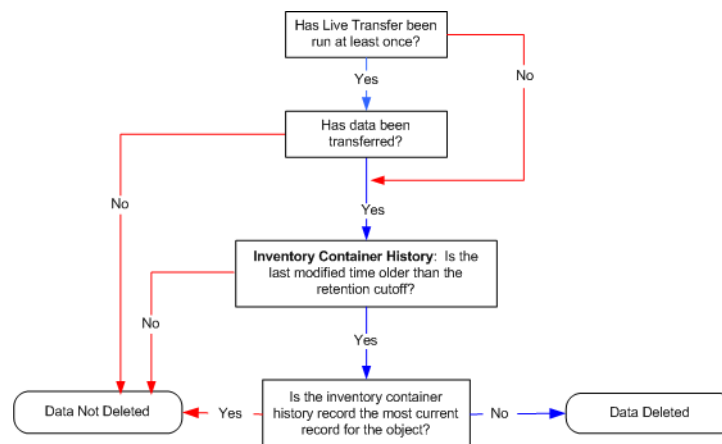
- ▶ If the object is a tracked object:
 - ❖ Does the object have the correct object status?
 - ❖ Does it meet the basic, group, consumption, and association purge requirements?

If the answer to both questions is yes, then the object is purged. All of these criteria must be met for the object to be purged.

Figure 3-11: Purging Tracked Objects

- If the object is an inventory container history:
 - ❖ Is the object's last modified time older than the data retention period?
 - ❖ Is it not the most current history record for the object?

If the answer to both questions is yes, then the object is purged. All of these criteria must be met for the object to be purged.

Figure 3-12: Purging Inventory Container Histories

The same rules used in purging the Production database apply to purging records from the ODS.

DCS Relationships

A DC_<objectName> table for non-static data collection sets (DCSs) can contain both standalone and non-standalone DCinstances. Whenever a status-based Purge is run, the marker scans for the standalone and non-standalone instances differently:

- ❑ All DC_<objectName> tables for non-static DCSs are scanned for standalone instances.
- ❑ Only a subset of DC_<objectName> runtime tables are scanned for non-standalone instances. This set of tables is determined by the DCS relationships stored in PURGE_DCS_MAP table.

When a status-based Purge is run for the first time, DCS relationships are checked automatically and the discovered relationships are stored in PURGE_DCS_MAP table after updating the DCS relationships. A DCS has a relationship with a type of object if runtime instances of the DCS were collected against that type of object. The DCS relationships speed the Purge process because Purge does not need to scan all the DC runtime tables every time for non-standalone instances. It only scans those DC runtime tables with a relationship.

For example:

1. An initial status-based Purge is run, and the following DCInstance:tracked object relationships are discovered:
 - ▶ DC_A: unit, lot, order
 - ▶ DC_B: unit, order
 - ▶ DC_C: lot, part
2. When a Purge job is run, the marker process marks a unit with unit_key = 1000 for purging.
3. The marker process scans DC_A and DC_B for DCInstances whose object_key = 1000 (associated to the unit) and marks them for purging.
4. The marker process does not scan DC_C because DC_C does not collect DCInstances for units.

It is important to regularly update the DCS relationships so that objects that need to be purged will be marked correctly. After you have run a status-based Purge for the first time, you should update your DCS relationships before each purge. The DCS Relationship Identifier is the process responsible for finding the types of objects that each DCS can collect against.

See “Updating Data Collection Maps (Status-Based Only)” on page 186 for more details.

Object Purge Criteria

When running status-based purge, different objects have different criteria for purging. Go to the following topics for the purge criteria used for each object type:

- ❑ “Batches” on page 45
- ❑ “ObjectStateHistory” on page 47
- ❑ “Process Orders” on page 47

- ❑ “Reports and Labels” on page 48
- ❑ “Stateless Runtime Objects” on page 49
- ❑ “Inventory Container History” on page 50
- ❑ “Tracked Runtime Objects” on page 51
- ❑ “Workflows” on page 55
- ❑ “Transactions” on page 55
- ❑ “Pallets” on page 56

Batches

IMPORTANT: Batches can only be marked for purge by using the ODS marker.

When a batch is purged, the batch along with the sublots and subplotRelation objects owned by that batch will be purged. For a batch to be purged, the purgeability requirements for all these objects must be met.

Batch objects are purged based on the following rules:

- ❑ **Basic:** purgeability is determined by last modified time and retention period.
- ❑ **Ownership Group:** purgeability is determined by the purgeability of all the members within the groups that the object belongs to.
- ❑ **Containment:** purgeability is determined by the purgeability of the batch’s container.
- ❑ **Consumption:** if a subplot participates in consumption, purgeability is determined by the purgeability of all of the object’s consuming objects. If the object does not participate in consumption, this component is automatically true.
- ❑ **Association:** the object’s association with workflows, application table instances, or references through named User-Defined Attributes (UDAs) or State Transition Attributes (STAs) affects its purgeability. All batches referred to through subplot relations must meet the purge criteria.

Basic Purge Rules

Before a batch is purged, the marker looks at its current state(s) and whether its last modified time is older than the retention period. If all of the flexible state model (FSM) states that the batch is currently on have the *sys.allow.Batch.purge* semantic property assigned to them, then the batch is in a purgeable state. If the batch is not on any FSMs, it is treated as though the semantic property was present.

Ownership Group Rules

Batches participate in ownership grouping by owning or being owned by other objects. For example, a batch forms an ownership group by owning sublots and subplotRelations. When you purge batches that are grouped in an ownership group, keep in mind the following:

- ❑ **ONE:** A group of objects must all have the same purgeability. A group is purgeable only if all its member objects are purgeable. For example:
 - ▶ A batch includes a subplot.
 - ▶ The subplot has a named UDA that points to an order step.
 - ▶ The order step is in a purgeable process order ownership group.
 - ▶ If the subplot cannot be purged, then the order step is not purgeable either, and neither is the entire batch and all of its child objects.
- ❑ **TWO:** When the marker determines the purgeability of the groups, the known purgeability of a group can propagate to other groups through common members. For example:
 - ▶ Batch A includes two sublots, Sublot1 and Sublot2.
 - ▶ Batch B includes one subplot, Sublot3, that contributes to Sublot2.
 - ▶ Batch B is not purgeable. Therefore, neither is Sublot3 or Sublot2.
 - ▶ If Sublot2 is not purgeable, then Batch A and Sublot1 not purgeable either.

Containment Purge Rules

Because carriers are buildtime objects, sublots contained in them cannot be purged.

Consumption Purge Rules

When purging consumed sublots, the consuming unit or lot is checked by the marker. You cannot purge a consumed object if any of its consuming objects cannot be purged.

Association Purge Rules

A batch's association with the following objects affects its purgeability:

- ❑ **Workflows:** If a batch is associated with a workflow that is not purgeable, the batch is not purgeable.
- ❑ **Application Table Instances:** A batch can be associated with application table instances through the object columns. If a batch is associated with an application table instance that is not purgeable, the batch is not purgeable.
- ❑ **References through Named UDAs or STAs:** If a batch is referenced by an unpurgeable object through a named UDA or STA, then the batch is not purgeable.

ObjectStateHistory

IMPORTANT: ObjectStateHistory records can only be marked for purge with an ODS marker.

ObjectStateHistory objects are purged based on the following rules:

- ☐ **Runtime objects:** if a runtime object is purgeable, all of its state history records are purgeable.
- ☐ **Buildtime objects:** if a buildtime object's state history record's exit time satisfies the retention period, that object is purgeable as long as it is not the current history record for the buildtime object.

Process Orders

IMPORTANT: Process orders can only be marked for purging by using the ODS marker.

When a process order is purged, the process order along with the following objects owned by that process order will be purged.

- ☐ PlannedConsumption
- ☐ PlannedProduction
- ☐ ProcessOrderItem
- ☐ ProcessStep
- ☐ ProcessStepConsumedSublotInfo
- ☐ ProcessStepProducedSublotInfo
- ☐ ControlRecipe
- ☐ OrderStep
- ☐ OrderStepInput
- ☐ OrderStepOutput
- ☐ ProducedSubLotInfo

For a process order to be purged, the purgeability requirements for all of the objects listed above must be met.

Process orders are purged based on the following rules:

- ☐ **Basic:** purgeability is determined by state, last modified time, and retention period.
- ☐ **Ownership Group:** purgeability is determined by the purgeability of all the members within the groups that the object belongs to.

- ❑ **Association:** the object's association with workflows, application table instances, or references through named UDAs and STAs affects its purgeability.

Basic Purge Rules

Before a process order is purged, the marker looks at its current state(s) and whether its last modified time is older than the retention period. If all of the FSM states that the process order is currently on have the *sys.allow.ProcessOrder.purge* semantic property assigned to them, then the process order is in a purgeable state. If the process order is not on any FSMs, it is treated as though the semantic property was present.

Ownership Group Rules

Process orders participate in ownership grouping by owning or being owned by other objects. For example, a process order forms an ownership group by owning process order items and control recipes. When you Purge process orders that are grouped in an ownership group, keep in mind that a group of objects must all have the same purgeability. A group is purgeable only if all its member objects are purgeable. For example:

- ❑ A subplot includes a named UDA that points to an order step.
- ❑ The order step is in a purgeable process order ownership group.
- ❑ If the subplot cannot be purged, then the order step is not purgeable either, and neither is the entire process order and all of its child objects.

Association Purge Rules

A process order's association with the following objects affects its purgeability:

- ❑ **Workflows:** If a process order is associated with a workflow that is not purgeable, the process order is not purgeable.
- ❑ **Application Table Instances:** Process orders can be associated with application table instances through the object columns. If a process order is associated with an application table instance that is not purgeable, the process order is not purgeable.
- ❑ **References through Named UDAs and STAs:** If a process order is referenced by a non-purgeable object through a Named UDA or STA, the process order is not purgeable.

Reports and Labels

IMPORTANT: Reports, labels, and labelHistory objects can only be marked for purging by the ODS marker.

Reports and labels are purged based on a retention period specified in days and their last modified time. LabelHistory objects are purged only when the owning Label is purged.

Stateless Runtime Objects

Stateless runtime objects are those objects created by Plant Operations at runtime that do not have a tracked status. They include non-static standalone DC instances, named UDA instances associated with stateless runtime objects, and any other objects that do not have a tracked status or are not associated with a tracked object. They are marked using a time-based approach that makes use of the retention period and the purge window (if being marked with the Production marker). If you are using the ODS marker, they are marked using only the retention period.

IMPORTANT: The retention period for stateless runtime objects can affect how tracked runtime objects are purged. The retention period for tracked runtime objects must be equal to or longer than the retention periods of stateless runtime objects and workflows. If this requirement is not met, the following error will result when the system attempts to purge tracked runtime objects: "Tracked object marker cannot run ahead of the stateless and workflow markers."

For every marked stateless runtime object, associated rows in the named UDA, STA, DC, and AT runtime tables (where applicable) are also marked. The following is a list of these objects:

- ❑ **DS_PERSISTENT_OBJECT:** This table does not have a site number. All rows whose auto-delete flag is set to "1" and were updated within the purge window are marked if you are using the Production marker.
- ❑ **RESOURCE_OBJECT_HISTORY:** Entries in this table unrelated to workflows are marked using the purge window if you are using the Production marker.
- ❑ **Unused INSTRUCTION_LIST:** A row in this table is marked if its INST_LIST_KEY key does not appear in the following tables:

• BOM	• PRODUCTION_LINE	• TBOM
• DCS	• PRODUCTION_QUEUE	• TEST_DEFINITION
• DCS_PARM	• RESOURCE_CONDITION	• TMPL_TBOM
• EQUIPMENT	• RESOURCE_ROUTE	• UNIT
• LOT	• ROUTE	• WORK_CENTER
• OPERATION	• ROUTE_STEP	• WORK_ORDER
• PART	• STATION	
- ❑ **Unused REFERENCE_DESIGNATOR:** A row in this table is marked if all of the following are met:

- ▶ REFERENCE_DESIGNATOR_KEY key does not appear in BOM_PART_LIST.
- ▶ REFERENCE_DESIGNATOR_KEY key does not appear in TBOM_PART_LIST.
- ❑ Unused NOTE_BLOB: A row in this table is marked if their NOTE_KEY key does not appear in the following tables:
 - ▶ LOT
 - ▶ UNIT
 - ▶ WORK_ORDER
 - ▶ WORK_ORDER_ITEMS
- ❑ Expired TMPL_TBOM, TMPL_TBOM_PART_LIST, TMPL_PART_LIST, TMPL_TBOM_SUBSET: The set of HANDLE_KEY keys of the TMPL_TBOM table is selected for rows that are of type “Top” and whose expiration time is equal to or more than 24 hours before the current time.
- ❑ Non-static standalone DC instances: First, the DCS table is scanned for definitions with static_data = 0. Second, the runtime table of these DCS definitions are scanned for rows that are within the purge window (if you are using the Production marker) and with the object_key = -1. Instances of DCS whose static_data = 0 and object_key = -1 are examined and marked accordingly. Keep in mind the following:
 - ▶ DC instances with an object_key not equal to -1 are non-standalone DC instances. They can only be purged if their parent object is purged.
 - ▶ DC instances with a static_data value equal to 1 are static DC instances and are never purged.
 - ▶ BOM, BOM_PART_LIST of type “WorkOrder”: Only BOMs of type “WorkOrder” are marked. The set of marked BOM_KEY keys are then used to mark the BOM_PART_LIST table by the BOM_KEY key.

Inventory Container History

IMPORTANT: Inventory Container History records can only be marked for purge with an ODS marker.

Inventory container history records are purged based only on the last modified time in relation to the retention period set and whether or not it is the current inventory container history record for an object. An object’s current inventory container history record is never purged regardless of how old it is.

Tracked Runtime Objects

Tracked runtime objects have a tracked status record. There are five kinds of tracked objects:

- ☐ BOX
- ☐ UNIT
- ☐ LOT
- ☐ WORK_ORDER
- ☐ WORK_ORDER_ITEM

Purging tracked objects depends on the following set of rules:

- ☐ **Basic:** If you are using the Production marker to determine purgeability, candidacy is determined by the tracked status record and the object's last modification time (given by the *xfr_update_pid*) with respect to the purge window. The ODS marker uses the same considerations but does not consider a purge window.
- ☐ **Group:** purgeability is determined by the purgeability of all the members within the groups that the object belongs to.
- ☐ **Consumption:** if the object participates in consumption, purgeability is determined by the purgeability of all of the object's consuming objects. If the object does not participate in consumption, this component is automatically true.
- ☐ **Association:** the tracked object's association with workflows and application table instances affects its purgeability.

For an object to be purged, the purgeability requirements for all of the levels listed must be met.

IMPORTANT: The retention period for tracked runtime objects must be equal to or longer than the retention periods of stateless runtime objects and workflows. If this requirement is not met, the following error will result when the system attempts to purge tracked runtime objects: "Tracked object marker cannot run ahead of the stateless and workflow markers."

Basic Purge Rules

Before a tracked object is purged, the marker looks at its status level and whether the object's last modified time falls within the purge window.

There are three status levels that apply to status-based purge:

- ☐ **Finished:** this is the lowest level. An object can be finished without being shipped or closed. Therefore, an object that has the status Finished is considered less closed than an object that has the status Shipped or Closed.

- ❑ **Shipped:** an object cannot be shipped unless it is already finished. However, an object can be shipped without being closed. Therefore, an object that has the status Shipped is considered more closed than an object that has the status Finished but less closed than an object that has the status Closed.
- ❑ **Closed:** this is the highest level. An object must be both finished and shipped before it can be closed. Therefore, an object that has the status Closed is considered more closed than an object that has the status Finished or Shipped.

An object will be purged depending on its status compared to the status level defined by the **Status for Purge** property. See “[Creating a Purge Job](#)” on page 174 for more details.

Group Purge Rules

Tracked objects form the following types of groups:

- ❑ **Ownership Group:** Objects participate in ownership grouping by owning or being owned by other objects. For example, a work order forms an ownership group by owning work order items, lots, and units. Ownership grouping forms a tree structure. Each group has a top-level object that is not owned by other objects, such as the work orders, standalone lots, and standalone units.
- ❑ **Containment Group:** Objects participate in containment grouping by containing or being contained in other objects. For example, a box forms a containment group by containing lots, units, and other boxes. Containment grouping forms a tree structure. Each group has a top-level object that is the outermost container not contained in other containers.

A tracked object can participate in both types of grouping.

Ownership Groups

When you purge tracked objects that are grouped in an ownership group, keep in mind the following:

- ❑ **ONE:** A group of objects must all have the same purgeability. A group is purgeable only if all its member objects are purgeable. If a box contains a unit that cannot be purged, then the box itself also cannot be purged.
- ❑ **TWO:** When the marker determines the purgeability of the groups, the known purgeability of a group can propagate to other groups through common members. For example:
 - ▶ Object O1 is a member of groups A, B, and C.
 - ▶ Object O2 is a member of group B.
 - ▶ Object O2 is not purgeable. Therefore, group B and every object in it are not purgeable, including object O1.
 - ▶ Because object O1 is not purgeable, then groups A and C and all the objects in them are also not purgeable.

- ❑ **THREE:** Because the marker operates on groups, the purgeability of a group is equal to the purgeability of its top-level object because of how the middle-tier handles status propagation through ownership groups:
 - ▶ If an action is performed on an object, the status change propagates downward unconditionally.
 - ▶ If an action is performed on an object that makes its status more closed (for example, changing it from Shipped to Finished), the status change propagates upward if all siblings of the object have an equal or more closed status.
 - ▶ If an action is performed on an object that makes its status less closed, the status change propagates upward unconditionally.
 - ▶ When a status change happens to an object, its tracked status record will have the *xfr_update_pid* value updated to the current system PID value.
- ❑ **FOUR:** No objects of an ownership group can have a tracked status that is less closed than that of the top-level object of the group. This allows the marker to check only the tracked status of the top-level object to determine whether all objects of an ownership group have an appropriate status. However, if you are using the Production marker to check individual objects for purging candidacy, these objects still need to be checked for the purge window requirement. The ODS marker does not consider a purge window. Consider the following scenario:
 - ▶ The Production marker is being used to mark items for purging.
 - ▶ Purge status: Finished (The **Status for Purge** property is set to Finished.)
 - ▶ Top-level object: *xfr_update_pid* = 97, status = Finished
 - ▶ Some lower level object: *xfr_update_pid* = 200, status = Shipped
 - ▶ Current purge window: (95, 100]

The top-level object appears to be purgeable because it has an appropriate status and is within the purge window. However, the lower level object lies outside of the purge window, which makes it not purgeable. Therefore, the top-level object AND the whole ownership group become not purgeable. Furthermore, propagation of purgeability of groups makes other groups that have common members with this group not purgeable.

This scenario shows that the purge window requirement (of the object rows and their tracked status records) for every object is checked by the marker.

Containment Groups

When you purge tracked objects that are grouped in a containment group, keep in mind the following:

- ❑ **Carriers:** because these objects are build time objects, their contents cannot be purged.

- ❑ **Reusable Boxes:** because these objects work like carriers with the performance benefits of boxes, their contents cannot be purged. No Shipped transactions are performed on reusable boxes.
- ❑ **Tracked Reusable Boxes:** because these objects work like reusable boxes with a tracked status on the box, their contents cannot be purged.
- ❑ **Group Boxes:** the contents of these boxes can be purged because one single tracked status record is shared by all the contents.
- ❑ **Normal Boxes:** the contents of these boxes can be purged. The box assumes the tracked status record of the first contained object. If the contents have mixed statuses, the purgeability of every contained and recursively contained object must be evaluated.

Consumption Purge Rules

When purging consumed objects, the consuming object is checked by the marker. When purging consumed objects and consuming objects, keep in mind the following:

- ❑ You cannot purge a consumed object if any of its consuming objects cannot be purged.
- ❑ Units consumed by other objects automatically get a closed status, and they can be purged when their consuming objects are purged.
- ❑ When a lot is consumed, the consumed lot can remain open even though the consuming objects are closed. Therefore, the consuming objects can be purged (given other requirements are met) even when a consumed lot is not purgeable.

The purgeability of a consuming object affects the purgeability of all objects that it consumes (and the objects that they in turn consume). For example:

- ❑ LotA is consumed by LotB.
- ❑ LotB is consumed by LotC.
- ❑ LotA and LotB are both closed. LotC is not closed.
- ❑ LotA cannot be purged because even though LotB is closed, LotB's consuming object (LotC) cannot be purged.

Purging Lots

When purge examines a lot, it looks at the lot type in the LOT_FLOW_TRACKING table. The following describes how Purge treats the different types of lots:

- ❑ Normal and serialized lots are treated the same by purge.
- ❑ Unserialized and bulk lots are treated the same by purge.
- ❑ If the type is a flow lot, Purge examines all the partial lots belonging to the flow lot. If all of them are within the purge window and have an appropriate

status, then the flow lot can be purged. If the flow lot is a final candidate, the belonging partial lots' LOT_FLOW_TRACKING rows and status records will be purged with the flow lot.

Association Purge Rules

A tracked object's association with the following objects affects its purgeability:

- ❑ **Workflows:** If a tracked object is associated with a workflow that is not purgeable, the tracked object is not purgeable.
- ❑ **Application Table Instances:** Tracked objects can be associated with application table instances through the object columns. If a tracked object is associated with an application table instance that is not purgeable, the tracked object is not purgeable.

Workflows

Workflows have a resource object status. The purge status for workflows is Closed and cannot be configured by users. The retention period configured for workflows determines which workflows will be purged. Only workflows that have their status as Closed and have their status last modified within the purge window are marked.

IMPORTANT: The retention period for workflows can affect how tracked runtime objects are purged. The retention period for tracked runtime objects must be equal to or longer than the retention periods of stateless runtime objects and workflows. If this requirement is not met, the following error will result when the system attempts to purge tracked runtime objects: "Tracked object marker cannot run ahead of the stateless and workflow markers."

Transactions

Transactions are purged separately from their objects using a time-based approach based on the retention period. For each marked transaction in the TRX_BASE database table, the source_type of the marked row may cause corresponding rows to be deleted from one or more of the following satellite tables:

- ❑ TRX_BOX
- ❑ TRX_CARRIER
- ❑ TRX_CONSUMPTION
- ❑ TRX_GENEALOGY

When configuring retention periods, keep in mind that transactions are purged according to the retention period and not the status of the objects to which they refer. All transactions records are purged for all objects that meet the retention

period even if the objects are not in the status provided in the **Status for Purge** property defined when configuring the Purge job.

Pallets

IMPORTANT: Pallets can only be marked for purge by using the ODS marker.

When a pallet is purged, the pallet along with the sublots owned by that pallet will be purged if the subplot and its containing batch are purgeable. For a pallet to be purged, the purgeability requirements for all these objects must be met.

NOTE: Because sublots are always contained in batches, their purgeability is always dependent on their containing batches. See “[Batches](#)” on [page 45](#) for a batch’s purge criteria.

Pallet objects are purged based on the following rules:

- ☐ **Basic:** purgeability is determined by last modified time and retention period.
- ☐ **Ownership Group:** purgeability is determined by the purgeability of all the members within the groups that the object belongs to.
- ☐ **Containment:** purgeability is determined by the purgeability of the pallet’s container.
- ☐ **Consumption:** if a subplot participates in consumption, purgeability is determined by the purgeability of all of the object’s consuming objects. If the object does not participate in consumption, this component is automatically true.
- ☐ **Association:** the object’s association with workflows, application table instances, or references through named User-Defined Attributes (UDAs) or State Transition Attributes (STAs) affects its purgeability. All pallets referred to through subplot relations must meet the purge criteria.

Basic Purge Rules

Before a pallet is purged, the marker looks at its current state(s) and whether its last modified time is older than the retention period. If all of the flexible state model (FSM) states that the pallet is currently on have the *sys.allow.Pallet.purge* semantic property assigned to them, then the pallet is in a purgeable state. If the pallet is not on any FSMs, it is treated as though the semantic property was present.

Ownership Group Rules

Pallets participate in ownership grouping by owning or being owned by other pallets. For example, a pallet forms an ownership group by owning sublots and

other pallets. When you purge pallets that are grouped in an ownership group, keep in mind the following:

- ❑ **ONE:** A group of objects must all have the same purgeability. A group is purgeable only if all its member objects are purgeable. For example:
 - ▶ A pallet includes a subplot.
 - ▶ The subplot has a named UDA that points to an order step.
 - ▶ The order step is in a purgeable process order ownership group.
 - ▶ If the subplot cannot be purged, then the order step is not purgeable either, and neither is the pallet and all of its child objects.
- ❑ **TWO:** When the marker determines the purgeability of the groups, the known purgeability of a group can propagate to other groups through common members. For example:
 - ▶ Pallet A includes two sublots, Sublot1 and Sublot2.
 - ▶ Pallet B includes one subplot, Sublot3, that contributes to Sublot2.
 - ▶ Pallet B is not purgeable. Therefore, neither is Sublot3 or Sublot2.
 - ▶ If Sublot2 is not purgeable, then Pallet A and Sublot1 not purgeable either.

Containment Purge Rules

Because pallets are buildtime objects, sublots contained in them cannot be purged.

Consumption Purge Rules

When purging consumed sublots, the consuming object is checked by the marker. You cannot purge a consumed object if any of its consuming objects cannot be purged.

Association Purge Rules

A pallet's association with the following objects affects its purgeability:

- ❑ **Workflows:** If a pallet is associated with a workflow that is not purgeable, the pallet is not purgeable.
- ❑ **Application Table Instances:** A pallet can be associated with application table instances through the object columns. If a pallet is associated with an application table instance that is not purgeable, the pallet is not purgeable.
- ❑ **References through Named UDAs or STAs:** If a pallet is referenced by an unpurgeable object through a named UDA or STA, then the pallet is not purgeable.

Reverse the ATRows/Tracked Object Dependency

By default, when performing status-based purge, the purgeability of referenced objects is dependent on the purgeability of the associated ATRow. For example:

- ☐ An ATDefinition contains a column that is an object reference to a class (for example, Unit).
- ☐ The class referenced by the ATDefinition is subject to being purged.
- ☐ The referenced object (a unit) meets the status and time retention requirements to be purged.
- ☐ The ATRow for the referenced object is not purgeable (for example, it has not met the object retention period requirements for purging).

In this case, the referenced unit will not be purged because the ATRow that references it is not purgeable. Therefore, the marker process will not mark the unit for purging.

Using the **DeleteATRowsForClosedTrackedObjects** ODS Marker feature, you can reverse this relationship and make an ATRow purgeable if all of the following requirements are fulfilled:

- ☐ The referenced object is purgeable.
- ☐ The referenced object is one of the following:
 - ▶ Order
 - ▶ Order Item
 - ▶ Lot
 - ▶ Unit
- ☐ The marker process is being performed on the ODS.

When the **DeleteATRowsForClosedTrackedObjects** feature is enabled, it will change the way the ODS marker treats object references from ATRows to an object in the Unit, Lot, OrderItem, or Order class. If the target of the object reference points to a purgeable object, the ATRow will be added to the list of purge candidates regardless of the ATRow's purge status or its eligibility under the stateless runtime object retention period. The following lists the changes in behavior:

- ☐ **Disabled DeleteATRowsForClosedTrackedObjects Feature:** ATRows are purged when their purge status is true and the duration since the last modified time is greater than or equal to the configured stateless runtime object retention period.
- ☐ **Enabled DeleteATRowsForClosedTrackedObjects Feature:** ATRows that have an object reference to a unit, lot, order item, or order that is purgeable are also purgeable. The ATRows' purge status and retention period configuration are not taken into account.

Limitations

Please note the following limitations of the **DeleteATRowsForClosedTrackedObjects** feature:

- ❑ This feature only deals with references to units, lots, order items, and orders. For example, if an ATRow refers to a ConsumedPart and the ATRow is not set to be purgeable, the marker will prevent the deletion of the unit/lot that owns the ConsumedPart.
- ❑ This feature is only applicable when status-based purge is using the ODS marker.
- ❑ If the ATRow that makes the object reference to the tracked object is contained in an AT Table that itself is in an ATDefinition that is marked as static, the ATRow will not be deleted, and the object reference will prevent the deletion of the tracked object.

Enabling the DeleteATRowsForClosedTrackedObjects Feature

Enable the **DeleteATRowsForClosedTrackedObjects** feature by inserting a record into the SITE_CONFIG table in the production database where the data is to be deleted. The **item_name** of the record is *DeleteATRowsForClosedTrackedObjects*, and the **item_value** of the record is *true*.

Figure 3-13: DeleteATRowsForClosedTrackedObjects Item

Table - dbo.SITE_CONFIG Summary	
item_name	item_value
failedLoginAttemptCount	0
redo_retry	5
maxelapse_retry	100
minelapse_retry	25
logTrxLevel	1
logObjHistoryLevel	1
logObjRevisionLevel	0
clientTimeout	0
disallowUserDeletion	false
CloseLotWhenCompletelyConsumed	true
UnXFRRows	DataRepaired
PTR_19266	Success
PTR_26920	1
authorizationCacheTimeout	1800
accessPrivilegeRequired	false
defaultUserGroupForNewUser	PlantOpsOperator
DeleteATRowsForClosedTrackedObjects	true
* NULL	NULL

For example:

```
insert into SITE_CONFIG (item_name, item_value) values ('DeleteATRowsForClosedTrackedObjects', 'true')
```

After you have inserted the `DeleteATRowsForClosedTrackedObjects` record, configure and schedule your status-based purge jobs as usual making sure that you select the ODS marker to perform the marking process. See [Configure Status-Based Purge Tasks](#) for details.

If the `item_value` is set to false or a `DeleteATRowsForClosedTrackedObjects` record does not exist in the database, then the marker will perform its default behavior (i.e., the purgeability of referenced objects is dependent on the purgeability of the associated ATRow).

Review the following sections for sample scenarios.

Example 1 - ATRows that reference a unit

The column `atr_key` in the `AT_Example` table references a unit.

Table 3-1 ATRows that Reference a Unit

<code>atr_key</code>	Referenced unit is purgeable?	ATRow is purgeable according to purge status and retention period rules?
1	No	No
2	No	Yes
3	Yes	No
4	Yes	Yes

When the **DeleteATRowsForClosedTrackedObjects** feature is disabled or not being used:

- ☐ The ODS marker only looks at the ATRow's purge status and retention period.
- ☐ The ATRows with keys 2 and 4 are identified as purge candidates.
 - ▶ For key 2, the ATRow is purged, but the referenced unit is not.
 - ▶ For key 4, both the ATRow and the referenced unit are purged.
- ☐ Because the ATRow with key 3 is not purgeable, the unit is removed from the candidate pool even though it has been deemed purgeable. The unit will not be deleted.
- ☐ Neither the unit nor the ATRow with key 1 are purgeable. Therefore, the ATRow and associated unit are not purged.

When the **DeleteATRowsForClosedTrackedObjects** feature is enabled:

- ☐ The marker generates a query that returns a list of ATRows that meet the purge status and retention period requirements and ATRows that reference units that are purgeable. This query returns the ATRows with keys 2, 3, and 4.
- ☐ The marker then generates "back references" from the unit to each of these ATRows. If the unit is either not a candidate to begin with or is removed from the candidate pool due to some other constraint (for example, a unit is in an open box), the referenced ATRow is removed from the candidate pool as well.

Therefore, the ATRow with key 2 is removed from the candidate pool because the referenced unit is not purgeable.

- ❑ Using the above queries, the maker determines that the ATRows with keys 3 and 4 will be deleted.

Example 2 - ATRows that reference an order

The DeleteATRowsForClosedTrackedObjects feature is enabled, and the ATRow references an order.

Table 3-2

Case	Is the order closed and its retention period satisfied?	Has the order been eliminated from the candidate pool due to a reason other than an ATRow reference?	Is the ATRow purge status true and its retention period satisfied?	Is the order purged?	Is the ATRow purged?
1	No	N/A	No	No	No
2	No	N/A	Yes	No	No
3	Yes	No	No	Yes	Yes
4	Yes	No	Yes	Yes	Yes
5	Yes	Yes	No	No	No
6	Yes	Yes	Yes	No	No

Case 1: Neither the order nor the ATRow are candidates because neither one has fulfilled the purge status and retention rules requirements. Therefore, neither is purged.

Case 2: The order is not a candidate, but the ATRow is a candidate because it has fulfilled the purge status and retention rules requirements. Because the ATRow is referencing an order that is not purgeable, the ATRow is removed from the candidate pool. Neither the order nor the ATRow is purged.

Cases 3 and 4: The order is a purge candidate, so the ATRow is also loaded as a candidate regardless of its purgeability status. Because the order is not removed from the candidate pool due to other constraints, both the order and the ATRow are deleted.

Case 5: The order is initially loaded into the candidate pool, so the ATRow is also loaded into the candidate pool. However, the order is then removed from the pool due to some other constraint (for example, a unit that belongs to the order is in an open box). On a following iteration of the candidate reduction phase, the marker detects the reference from the order to the ATRow and that the order is no longer a candidate. Because of this, it will remove the ATRow from the candidate pool as well. Neither the order nor the ATRow is purged.

Case 6: This case is very similar to case 5. After the order is removed from the candidate pool, the ATRow is removed from the candidate pool as well due to the reference from the order. Neither the order nor the ATRow is purged.

Example 3 - ATRows that reference more than one instance of an order

The DeleteATRowsForClosedTrackedObjects feature is enabled, and the ATRows reference more than one instance of an order.

Table 3-3

Case	Is Order1 closed and its retention period satisfied?	Has Order1 been eliminated from the candidate pool due to a reason other than an ATRow reference?	Is Order2 closed and its retention period satisfied?	Has Order2 been eliminated from the candidate pool due to a reason other than an ATRow reference?	Is the ATRow's purge status true and its retention period satisfied?	Is Order1 purged?	Is Order2 purged?	Is the ATRow purged?
1	No	N/A	No	N/A	No	No	No	No
2	No	N/A	No	N/A	Yes	No	No	No
3	No	N/A	Yes	No	No	No	No	No
4	No	N/A	Yes	No	Yes	No	No	No
5	No	N/A	Yes	Yes	No	No	No	No
6	No	N/A	Yes	Yes	Yes	No	No	No
7	No	N/A	No	N/A	No	No	No	No
8	No	N/A	No	N/A	Yes	No	No	No
9	Yes	N/A	No	N/A	No	No	No	No
10	Yes	No	No	N/A	Yes	No	No	No
11	Yes	No	Yes	No	No	Yes	Yes	Yes
12	Yes	No	Yes	No	Yes	Yes	Yes	Yes
13	Yes	No	Yes	Yes	No	No	No	No
14	Yes	No	Yes	Yes	Yes	No	No	No
15	Yes	Yes	No	N/A	No	No	No	No
16	Yes	Yes	No	N/A	Yes	No	No	No
17	Yes	Yes	Yes	No	No	No	No	No
18	Yes	Yes	Yes	No	Yes	No	No	No

Table 3-3

Case	Is Order1 closed and its retention period satisfied?	Has Order1 been eliminated from the candidate pool due to a reason other than an ATRow reference?	Is Order2 closed and its retention period satisfied?	Has Order2 been eliminated from the candidate pool due to a reason other than an ATRow reference?	Is the ATRow's purge status true and its retention period satisfied?	Is Order1 purged?	Is Order2 purged?	Is the ATRow purged?
19	Yes	Yes	Yes	Yes	No	No	No	No
20	Yes	Yes	Yes	Yes	Yes	No	No	No

In this situation, the ATRow is only purgeable if both of the referenced orders are also purgeable. The ATRow will not be purged if only one referenced order is purgeable. Therefore, the only cases when all three objects are purged are cases 11 and 12.

- ☐ In case 11, the ATRow is not a candidate, but because both orders that the ATRow references are candidates, all three objects are purged.
- ☐ In case 12, all three objects are candidates, so all three are purged.

In all the other cases, the three objects are not purgeable because at least one of the following conditions is true:

- ☐ At least one order has not fulfilled the purge status and retention rules requirements. This applies to cases 1-10 and 15-16.
- ☐ At least one order has fulfilled the purge status and retention rules requirements but was then removed from the candidate pool due to some other constraint (for example, a unit that belongs to the order is in an open box). This applies to cases 13-20.

SQL Throttle

The SQL Throttle is used to minimize the impact of running Purge on the Production database by restricting the number of SQL statements that can be issued and executed by the Purge processes within any given period of time. Without the throttle, the Purge processes may issue many SQL statements in a short period of time and significantly reduce the database server's bandwidth to serve the production processes. For example, the reaper, which executes DELETE statements in a tight loop, can issue hundreds of SQL statements per second and will tie down the database server if left uncontrolled.

When using the SQL Throttle, users specify the minimum amount of time in milliseconds that must pass between the execution of SQL statements from the

Purge processes. After a statement is executed (This could be a batch statement, but the whole batch is considered one execution), the throttle takes a timestamp. This current time delayed by the minimum wait time (the **Throttle** property) will be the earliest time when the next execution request from Purge is allowed to proceed. If a request comes in before that time, the requesting thread is forced to sleep until the time is reached. The timestamp is kept at the database agent level, which means that if two processes create two distinct instances of the agent, they will be governed by two instances of the throttle.

When you change the **Throttle** property, the changes take effect when either the Purge job is restarted or when the next Purge job is run.

The SQL Throttle affects only the Purge processes (marker and reaper) and does not affect any other Data Management or Plant Operations processes.

Installing and Running Live Transfer

In this chapter

- ❑ **Installing Live Transfer 66**
 - Installing Live Transfer on Linux 69
- ❑ **Configuring Live Transfer Installations 71**
 - Defining the BLOB Payload Size (Optional) 71
 - Configuring Continuous Live Transfer (Optional) 72
 - Increasing the Message Resend Interval (Optional) 72
- ❑ **Registering and Starting Live Transfer Components 73**
 - Windows 73
 - Linux 74
- ❑ **Stopping and Unregistering Live Transfer Components 76**
- ❑ **Viewing Live Transfer Logs 77**

When you access FTPC Administrator, a summons is sent out to find all Live Transfer components listening on the `FTPC_ADMIN_TOPICS` topic in ActiveMQ. FTPC Administrator waits approximately 10 seconds for responses from unconfigured and configured components. If the component is configured, its configuration is retrieved and stored on the FTPC Administrator server's local disk. Because of this summons, the Live Transfer components should be installed and running before you try to launch FTPC Administrator.

TIP: If you want to send the summons again without logging out and logging back into FTPC Administrator, click the *Search Purge Services* link. This will summon the Live Transfer components as well as search for Purge services.

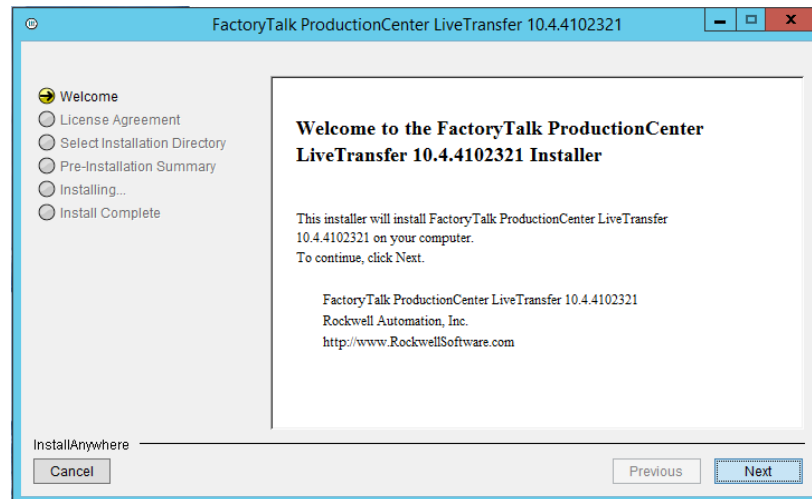
Perform the steps in the following sections to install and run Live Transfer.

Installing Live Transfer

You must install Live Transfer on every machine that is running a component of Live Transfer. For example, if you have the LiveTransferExtractor components running on one physical machine and the TLM running on another physical machine, Live Transfer must be installed on each machine. If your external ActiveMQ is running on a third machine, Live Transfer must be installed on that machine as well.

IMPORTANT: The following steps should be performed on a Windows client machine even if Live Transfer will be installed on a Linux machine.

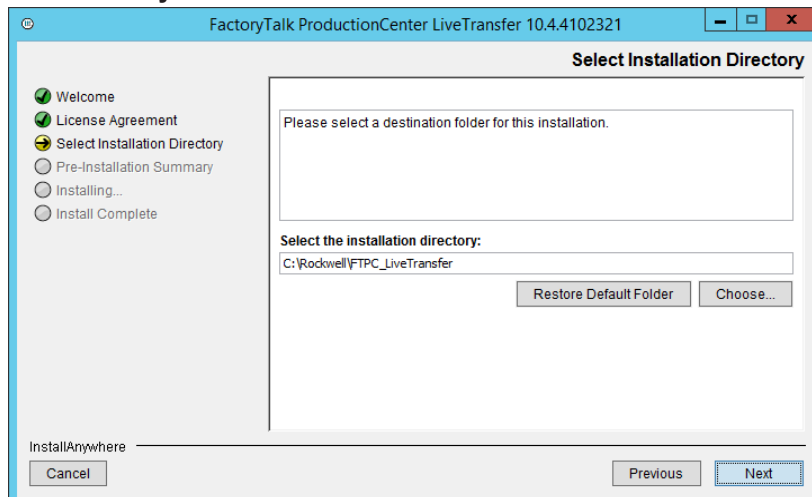
1. After downloading it from the Rockwell Software site, double-click the Live Transfer executable file.
2. After the wizard initializes the setup, a Welcome screen appears. Click [Next] to proceed.

Figure 4-1: Welcome Screen

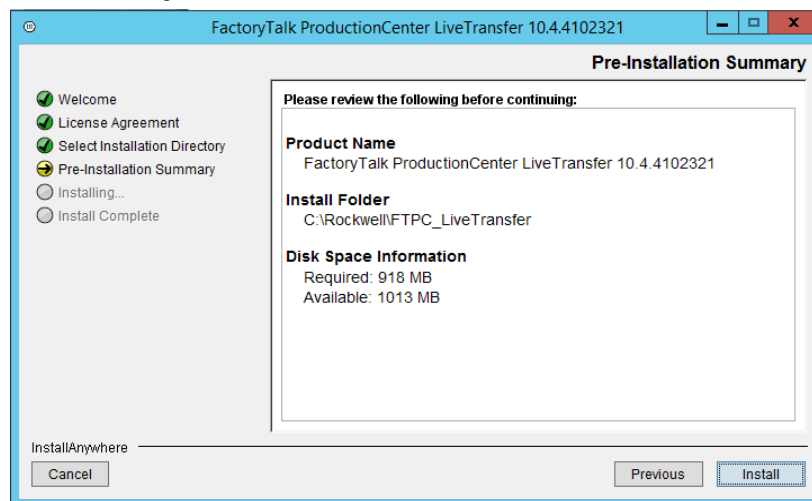
3. On the License Agreement screen, accept the license agreement, and then click [Next].

Figure 4-2: License Agreement Screen

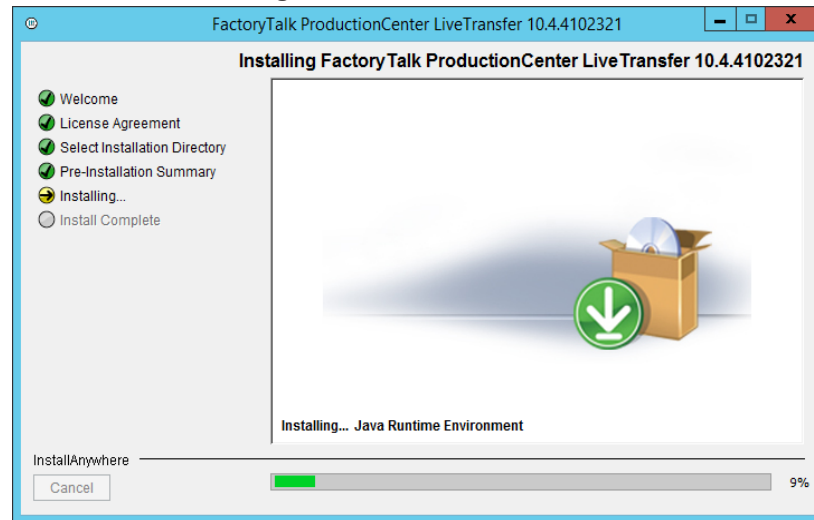
4. The Directory Selection screen allows you to extract the Live Transfer files to any drive or in any directory you want. The installer provides a default location of C:\Rockwell\FTPC_LiveTransfer. Perform one of the following, then click [Next].
 - Accept the default location.
 - Browse to the directory where you would like to extract the Live Transfer files. If you enter a path that does not exist, the folders will be created.
 Note the location because the file location will be used in upcoming steps.

Figure 4-3: Directory Selection Screen

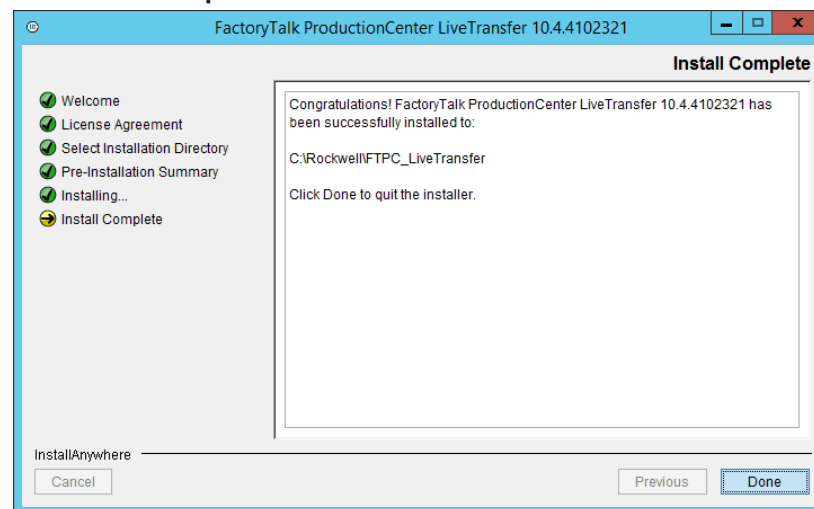
5. The Summary screen confirms the directory where the Live Transfer files will be placed. Verify that this information is correct, and then click [Install]. If the information is not correct, click [Previous] to return to the previous screen and correct the information.

Figure 4-4: Summary Screen

The installer displays a progress screen as Live Transfer is installing.

Figure 4-5: Extraction in Progress Screen

6. When all the files have been placed on the hard drive, an Install Complete screen appears. Click [Done] to exit the program.

Figure 4-6: Install Complete Screen

IMPORTANT: When installing Live Transfer, the internal and external ActiveMQ are installed for you. If you currently have any instances of ActiveMQ already installed, make sure that they are not using any ports that are being used by Live Transfer. By default, Live Transfer uses ports 61615 and 61616. See [“Creating Live Transfer Instances”](#) on page 146 for details.

Installing Live Transfer on Linux

If you are installing Live Transfer on a Linux machine, continue the installation by performing the following steps:

1. Copy the installation folder that was created when you performed the steps in “Installing Live Transfer” on page 66 to your Linux machine.
2. Log in to the Linux machine as root user.
3. Install the required J2SDK, accepting all defaults.
4. Open a shell, and enter the following command: **#vi /root/.bashrc**
5. Add the following lines to this file. If the entry exists, find and modify it.


```
export JAVA_HOME=<java_install_directory>
export PATH=$JAVA_HOME/bin:$PATH
```
6. Save and exit from the vi editor.
7. From a shell, enter the following: **#source /root/.bashrc**
8. Enter the following from the command line to verify the Java version available:


```
java -version
```

The resulting version should be the version of J2SDK you installed just now. If not, you may need to check your work and run the command again.
9. Uncompress the ActiveMQ TAR file by changing to the `<LT_install>/linux/JMS-Broker` directory and running the following command:


```
tar xvf apache-activemq-<version>-bin.tar
```
10. Convert the file formats to a Linux file format by calling the following command in the listed directories:

IMPORTANT: If you are on Linux 7.0, you must manually install the **dos2unix** package before running the following commands. The rpm package is located in the Packages directory of your installed Linux iso file.

```
<LT_install>/linux/LiveTransferExtractor/conf> dos2unix *.*
<LT_install>/linux/LiveTransferExtractor/bin> dos2unix *.*
<LT_install>/linux/TableLoaderManager/conf> dos2unix *.*
<LT_install>/linux/TableLoaderManager/bin> dos2unix *.*
```

11. Change the files' access privileges. When the files were copied over in [step 1](#), their access privileges were changed to Read Only by default. Change the access privileges to Read/Write/Execute by calling the following command in the listed directories:

```
<LT_install>/linux/LiveTransferExtractor/bin> chmod 777 *
<LT_install>/linux/TableLoaderManager/bin> chmod 777 *
<LT_install>/lib> chmod 777 *
```


Configuring Live Transfer Installations

If you have Live Transfer installed, perform the following steps before accessing FTPC Administrator.

1. Open the *AdmindedService.properties* file for the Live Transfer Extractor located at `<LT_install>\<win or linux>\LiveTransferExtractor\conf`. If you installed Live Transfer to the default location, the file is located at `C:\Rockwell\FTPC_LiveTransfer\<win or linux>\LiveTransferExtractor\conf` by default.
2. Set the `FTPC_ADMIN_JMS_URL` property to match the `FTPC_ADMIN_JMS_URL` property value defined in FTPC Administrator's *ftpcAdmin.properties* file located by default at one of the following locations:
 - ▶ **JBoss Standalone:**
`C:\Rockwell\PO<version>.<build>\FTPCAdminHome\conf`
 - ▶ **JBoss Advanced or WebSphere:** `C:\Program Files\Rockwell Software\FactoryTalk ProductionCentre\FTPCAdmin.war` under `WEB-INF\lib\`

IMPORTANT: If all of your components (FTPC Administrator, Live Transfer, Purge, application server, etc.) are using the same ActiveMQ instance, the JMS URL for each component must point to the same ActiveMQ instance regardless of where the component is installed.

3. Define the component's instance ID by setting the `SERVICE_ID` property. The ID for each component must be unique among all the Live Transfer, Loader, and Purge services.
4. Save and close the file.
5. Repeat [step 1](#) through [step 4](#) for the *AdmindedService.properties* file of the following Live Transfer services:
 - ▶ **TEM:** file located at `<LT_install>\<win or linux>\TableExtractManager\conf`.
 - ▶ **TLM:** file located at `<LT_install>\<win or linux>\TableLoaderManager\conf`.

Defining the BLOB Payload Size (Optional)

If you will be transferring a large amount of Binary Large Object (BLOB) data, define the **blobPayloadSize** property in the *SiteExtractManager.properties* file located at `<Live_Transfer_install>\<os>\SiteExtractManager\conf`.

```
blobPayloadSize=<size_in_bytes>
```

The default is 2097152, which is 2 MB.

- ❑ If the BLOB data is bigger than the defined `blobPayloadSize`, the file is split into multiple files that are packaged into a ZIP file by the extractor process and then loaded by the load process from the ZIP file. The default value is 2MB.
- ❑ If the `blobPayloadSize` is set to -1, the BLOB data is split into multiple files regardless of its size and then packaged into a ZIP file by the extractor process and then loaded by the load process from the ZIP file.

Configuring Continuous Live Transfer (Optional)

You can configure Live Transfer to run continuously by setting the **extractPeriod** property to its minimum value of 1. This property is defined in the *SiteExtractManager.properties* file located at `<Live_Transfer_Install>\<OS>\SiteExtractManager\conf`.

Because extract requests will not be sent until the current extract is completed (see “ActiveMQ Queue Throttling” on page 20 for details), setting the **extractPeriod** to 1 will ensure that a new request is sent immediately after the current extract has finished.

Increasing the Message Resend Interval (Optional)

If you want to increase the amount of time before messages are resent, perform the following steps one at a time and determine if it satisfies your requirements for resending before continuing to the next step. These properties are found in the *SiteExtractManager.properties* file located at `<Live_Transfer_Install>\<OS>\SiteExtractManager\conf`.

1. Decrease the value of the **numOfRowsPerMessage** property, which is 500 by default. With a smaller number of rows in each extract message, the loader can quickly send back an acknowledgment letting the Table Extractor know that the load is in progress.
2. Increase the value of the **reSendTimeout** property, which is 240000 milliseconds (four minutes) by default.
3. Decrease the value of the **maxUnacknowledgedMessages** property, which is 4 by default.

Registering and Starting Live Transfer Components

NOTE: Please note that Live Transfer and FTPC installed using the JBoss Standalone installer each install an instance of ActiveMQ that uses the same port by default (61616). If both these installations are running on one machine, you only need to have one of the ActiveMQ instances running. Make sure both instances are not trying to run at the same time, which can result in a port conflict.

Perform the following steps to register and start the Live Transfer components.

IMPORTANT: Before running Live Transfer, you must configure and run the *populatePdXfrUpdatePid.bat* file to completion after you have migrated your ODS. See “[Migrating a Base ODS](#)” on page 108 for more information on this file and migrating your ODS.

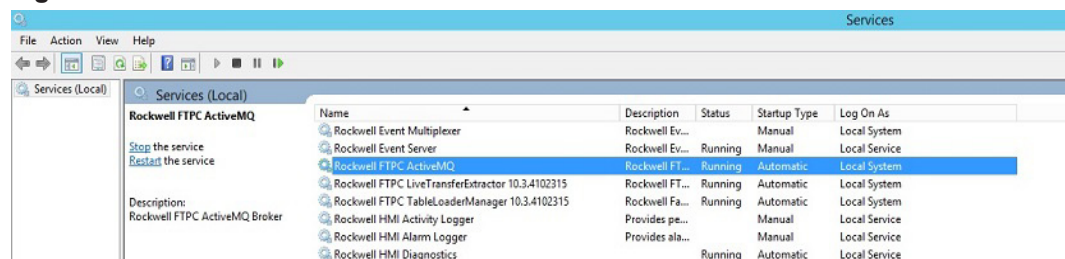
Windows

To register and start Live Transfer on a Windows machine, perform the following steps:

1. To register ActiveMQ as a service, go to Start > Programs > Rockwell Software > FTPC Purge <version> > ActiveMQ > Register Service. If you are using both Live Transfer and Purge, you only have to register ActiveMQ once.

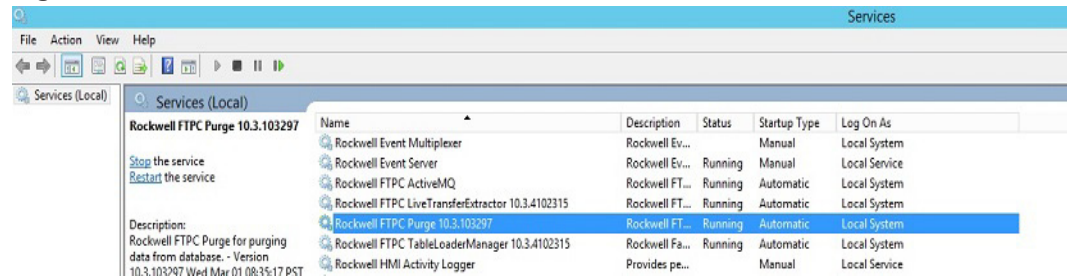
Once registered, the ActiveMQ service appears in the Services list.

Figure 4-7: ActiveMQ Service



2. Start ActiveMQ by going to Start > Programs > Rockwell Software > FTPC Purge <version> > ActiveMQ > Start Service.
3. Register the Live Transfer Extractor and Loader as services by clicking the following menu items:
 - ▶ Start > Programs > Rockwell Software > FTPC Live Transfer <version> > Live Transfer Extractor > Register Service
 - ▶ Start > Programs > Rockwell Software > FTPC Live Transfer <version> > Table Loader Manager > Register Service

Once they are registered, the components appear in the Services list.

Figure 4-8: Live Transfer Services

4. To start the Live Transfer Extractor and the Table Loader Manager, go to Start > Programs > Rockwell Software > FTPC Live Transfer <version>, and click the following menu items:
 - ❖ Live Transfer Extractor > Start Service
 - ❖ Table Loader Manager > Start Service

Linux

To register and start Live Transfer on a Linux machine, perform the following steps:

1. If you are on Linux 7.0, update the wrapper files by performing the steps below. If not, skip to [step 2](#):
 - a. Go to following website to download the 64-bit version wrapper for Linux: <http://wrapper.tanukisoftware.com/doc/english/download.jsp> .
 - b. Go to the *bin* folder of the wrapper installer you downloaded in [step a](#) and copy the wrapper file.
 - c. Replace the wrapper file in `<Live_Transfer_install>/linux/LiveTransferExtractor/bin` by pasting in the new wrapper file.
 - d. Replace the wrapper file in the following locations by repeating [step c](#):
 - `<Live_Transfer_install>/linux/SiteExtractManage/bin`
 - `<Live_Transfer_install>/linux/TableExtractManage/bin`
 - `<Live_Transfer_install>/linux/TableLoaderManager/bin`
 - e. Go to the *lib* folder of the wrapper installer you downloaded in [step a](#) and copy the wrapper.jar file.
 - f. Replace the wrapper.jar file in `<Live_Transfer_install>/lib` by pasting in the new wrapper.jar file.
2. To register ActiveMQ as a service that will automatically start on boot, run the following commands:

```
ln -s /rockwell/apache-activemq-<version>/bin/linux-x86-64/activemq
```

```
/etc/init.d/activemq
#chkconfig --add activemq
#chkconfig activemq on
```

If you are using both Live Transfer and Purge, you only have to register ActiveMQ once.

TIP: If you want to start ActiveMQ without registering it as a service, run the following command:

```
<Purge_install>/linux/JMS-Broker/apache-activemq-<version>/bin/activemq
```

3. To register Live Transfer Extractor as a service that will automatically start on boot, perform the following steps:

- a. Copy the runLiveTransferExtractor.sh file located at `<LT_install>/linux/LiveTransferExtractor/bin/` to the `etc/init.d` directory and rename it to `runExtractor`. Then, run the following commands:

```
#cd /rockwell/LiveTransfer/linux/LiveTransferExtractor/bin
#cp -a runLiveTransferExtractor.sh /etc/init.d/runExtractor
```

- b. Open `runExtractor` with a text editor and modify the `APP_PATH` parameter as shown:

```
APP_PATH="/rockwell/LiveTransfer/linux/LiveTransferExtractor"
```

- c. Run the following commands to register Live Transfer Extractor as a service that will automatically start on boot:

```
#chkconfig --add runExtractor
#chkconfig runExtractor on
```

4. To register Table Loader Manager as a service that will automatically start on boot, perform the following steps:

- a. Copy the `runLoader.sh` file located at `<LT_install>/linux/TableLoaderManager/bin/` to the `etc/init.d` directory and rename it to `runLoader`. Then, run the following commands:

```
#cd /rockwell/LiveTransfer/linux/TableLoaderManager/bin
#cp -a runLoader.sh /etc/init.d/runLoader
```

- b. Open `runLoader` with a text editor and modify the `APP_PATH` parameter as shown:

```
APP_PATH="/rockwell/LiveTransfer/linux/TableLoaderManager"
```

- c. Run the following commands to register Table Loader Manager as a service that will automatically start on boot:

```
#chkconfig --add runLoader
#chkconfig runLoader on
```

TIP: If you want to start Live Transfer Extractor and the Table Loader Manager without registering them as services, run the following commands:

```
<LT_install>/linux/LiveTransferExtractor/bin/runLiveTransferEx
tractor.sh start
```

```
<LT_install>/linux/TableLoaderManager/bin/runLoader.sh start
```

Once Live Transfer is started, it runs in the background automatically if data that needs to be transferred is found. The extractor checks to see if a new extract needs to be initiated at a 30 second interval by default.

IMPORTANT: If you need to delete an application table definition, a DCS definition, a named UDA definition, or an STA definition, you must stop Live Transfer first, delete the definition, and then restart Live Transfer.

Stopping and Unregistering Live Transfer Components

Perform the steps in the following sections to stop and unregister the Live Transfer components.

1. To stop Live Transfer, perform the following:

- ▶ **Windows:** stop the Live Transfer Extractor first and then stop the Table Loader Manager by going to the following.

- ❖ Start > Programs > Rockwell Software > FTPC Live Transfer <version> > Live Transfer Extractor > Stop Service
- ❖ Start > Programs > Rockwell Software > FTPC Live Transfer <version> > Table Loader Manager > Stop Service

- ▶ **Linux:** stop the Live Transfer Extractor first and then stop the Table Loader Manager using the following commands:

```
<LT_install>/linux/LiveTransferExtractor/bin/runLiveTransferE
xtractor.sh stop
```

```
<LT_install>/linux/TableLoaderManager/bin/runLoader.sh stop
```

2. To unregister Live Transfer, perform the following:

- ▶ **Windows:** go to Start > Programs > Rockwell Software > FTPC Live Transfer <version>:
- ❖ Live Transfer Extractor > Unregister Service
- ❖ Table Loader Manager > Unregister Service

- ▶ **Linux:** run the following commands:


```
#chkconfig --del runExtractor
#chkconfig --del runLoader
```
- 3. To stop ActiveMQ, perform the following:
 - ▶ **Windows:** Go to Start > Programs > Rockwell Software > FTPC Live Transfer <version> > ActiveMQ > Stop Service.
 - ▶ **Linux:** Press [Ctrl] + C.
- 4. To unregister ActiveMQ, perform the following:
 - ▶ **Windows:** go to Start > Programs > Rockwell Software > FTPC Purge <version> > ActiveMQ > Unregister Service.
 - ▶ **Linux:** run the following command:


```
#chkconfig --del activemq
```

Viewing Live Transfer Logs

Diagnostics are provided through log files that are generated by the separate components of Live Transfer. These logs are located in the following folders:

- ❑ Site Extract Manager and Table Extract Manager logs:

<LT_install>\<os>\LiveTransferExtractor\logs

- ❑ Table Loader Manager logs:

<LT_install>\<os>\TableLoaderManager\logs

where

<LT_install> is the location where you installed Live Transfer.

<os> is either **win** or **linux** depending on your operating system.

Each component of Live Transfer has two logs named <component_name>*.log and <component_name>_Error*.log where <component_name> is either SiteExtractManager, TableExtractManager, or TableLoaderManager. The error log only contains entries related to errors. All types of entries, including errors, are entered in the other log.

Once a log file reaches a certain size, the logging information is sent to a new log file, so multiple files will be present in this directory. To locate the most recent log file, look at the file with the latest timestamp.

A logging.properties file is provided that allows you to configure the logging output. See “[Configuring Live Transfer Logging](#)” on page 159 for details.

Installing and Running Purge

In this chapter

- ❑ **Installing Purge 80**
 - Installing Purge on Linux 83
- ❑ **Configuring Purge Installations 84**
- ❑ **Registering and Starting the Purge Service 85**
 - Windows 85
 - Linux 86
- ❑ **Stopping and Unregistering the Purge Service 88**
- ❑ **Viewing Purge Logs 89**

When you access the FTPC Administrator home page, you can click the **Search Purge Services** link to find any existing Purge services. In order for the Purge services to be returned, you must have Purge installed and running before accessing FTPC Administrator.

Perform the steps in the following sections to install and run Purge.

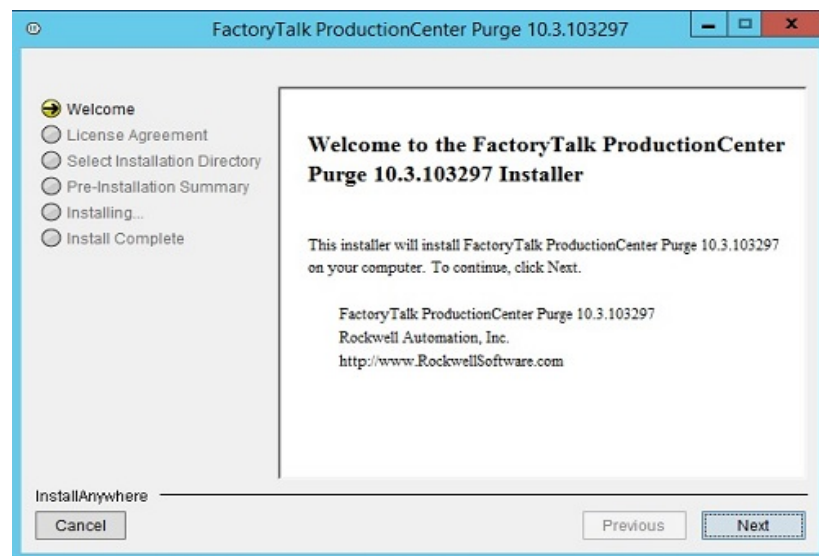
Installing Purge

You must install Purge on every machine that is running a component of Purge. For example, if your ActiveMQ is running on another machine, Purge must be installed on that machine as well.

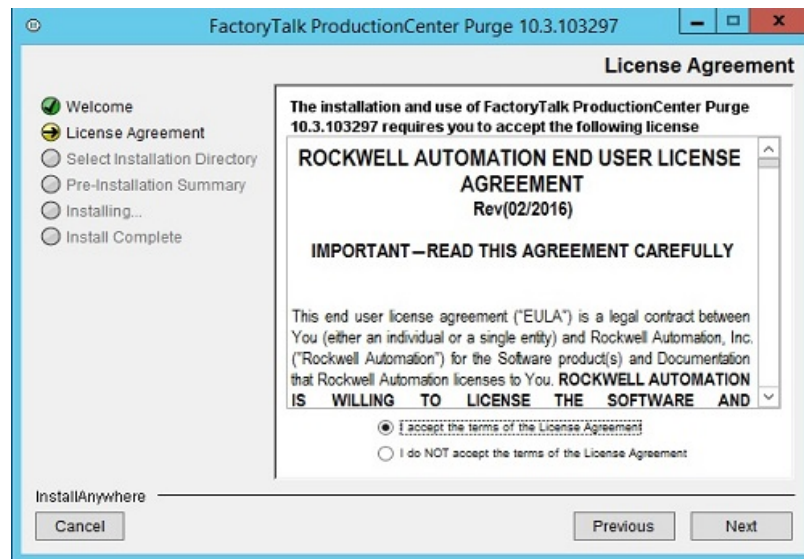
IMPORTANT: The following steps should be performed on a Windows client machine even if Purge will be installed on a Linux machine.

1. After downloading it from the Rockwell Software site, double-click the Purge executable file.
2. After the wizard initializes the setup, a Welcome screen appears. Click [Next] to proceed.

Figure 5-1: Welcome Screen

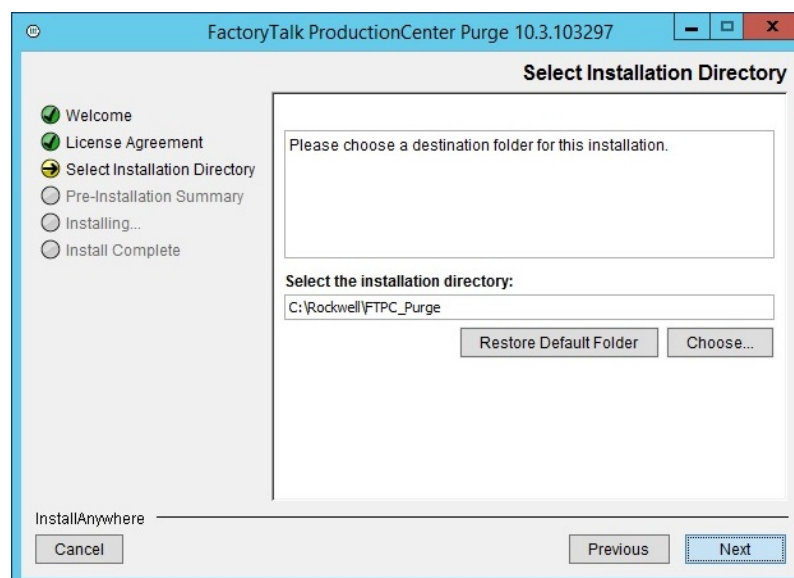


3. On the License Agreement screen, accept the license agreement, and then click [Next].

Figure 5-2: License Agreement Screen

4. The Directory Selection screen allows you to extract the Purge files to any drive or in any directory you want. The installer provides a default location of C:\Rockwell\FTPC_Purge. Perform one of the following, then click [Next].
 - ▶ Accept the default location.
 - ▶ Browse to the directory where you would like to extract the Purge files. If you enter a path that does not exist, the folders will be created.

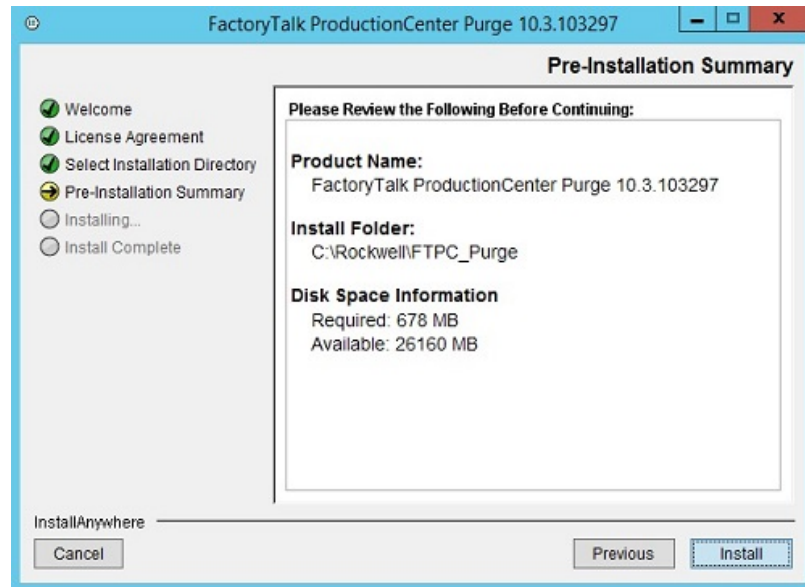
Note the location because the file location will be used in upcoming steps.

Figure 5-3: Directory Selection Screen

5. The Summary screen confirms the directory where the Purge files will be placed. Verify that this information is correct, and then click [Install]. If the

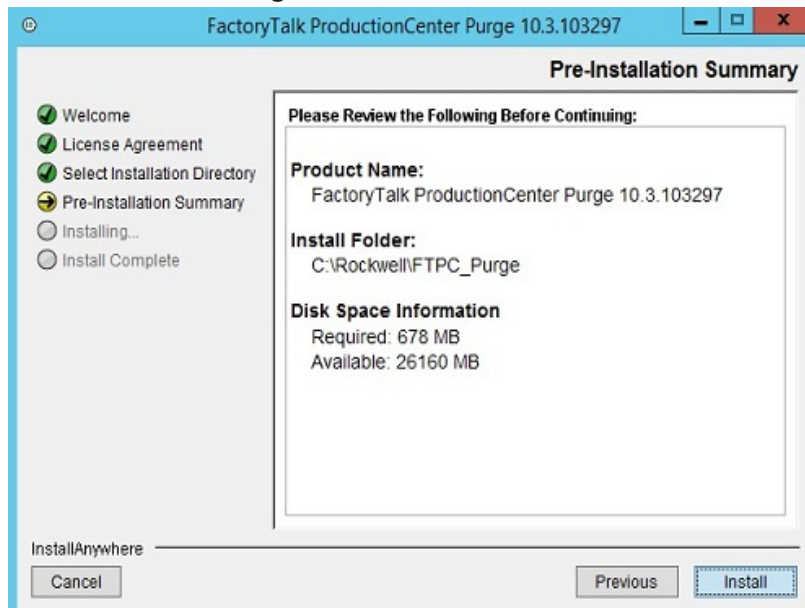
information is not correct, click [Previous] to return to the previous screen and correct the information.

Figure 5-4: Summary Screen

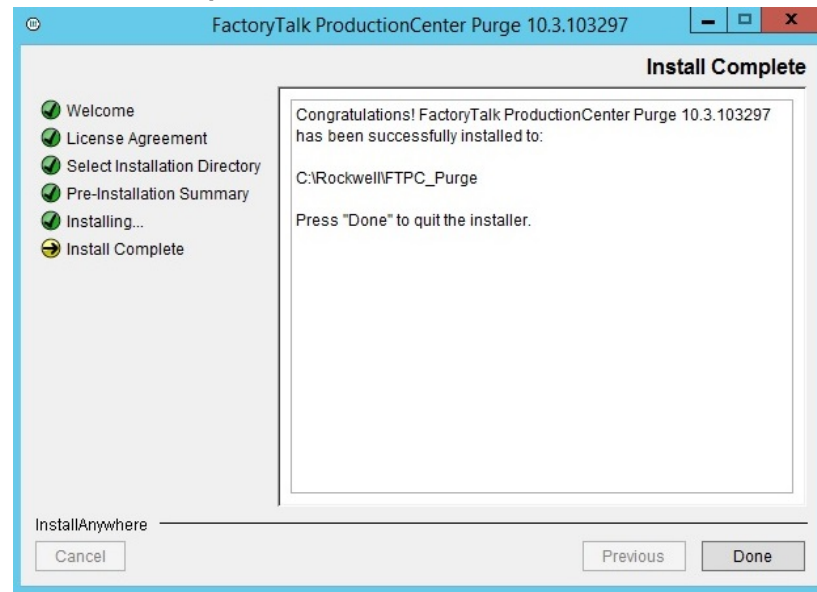


The installer displays a progress screen as Purge is installing.

Figure 5-5: Extraction in Progress Screen



6. When all the files have been placed on the hard drive, an Install Complete screen appears. Click [Done] to exit the program.

Figure 5-6: Install Complete Screen

When installing Purge, ActiveMQ is installed for you. If you currently have any instances of ActiveMQ already installed, you can configure the connection information for Purge on your ActiveMQ instance, and you will not need to use the ActiveMQ instance that is installed by the Purge installer. By default, Purge uses port 61616.

Installing Purge on Linux

If you are installing Purge on a Linux machine, continue the installation by performing the following steps:

1. Go to `<Purge_install>\linux\PurgeServices\conf` and open the `wrapper.conf` file.
2. Add the following property at the beginning of the file:


```
set.JAVA_HOME=<java_install_directory>
```

For example:

```
set.JAVA_HOME=/usr/java/jdk1.8.0_26
```

3. Save and close the file.
4. Copy the installation folder that was created when you performed the steps in “Installing Purge” on page 80 to your Linux machine.
5. Log in to the Linux machine as root user.
6. Install the required J2SDK, accepting all defaults.
7. Open a shell, and enter the following command:

```
#vi /root/.bashrc
```

8. Add the following lines to this file. If the entry exists, find and modify it.

```
export JAVA_HOME=<java_install_directory>
export PATH=$JAVA_HOME/bin:$PATH
```

9. Save and exit from the vi editor.
10. From a shell, enter the following:

```
#source /root/.bashrc
```

11. Enter the following from the command line to verify the Java version available:

```
java -version
```

The resulting version should be the version of J2SDK you installed just now. If not, you may need to check your work and run the command again.

12. Uncompress the ActiveMQ TAR file by changing to the `<Purge_install>/linux/JMS-Broker` directory and running the following command:

```
tar xvf apache-activemq-<version>-bin.tar
```

13. Convert the file formats to a Linux file format by calling the following command in the `<purge_install>/linux/PurgeServices/bin` directory:

```
dos2unix *.* *
```

IMPORTANT: If you are on Linux 7.0, you must manually install the `dos2unix` package before running this command. The `rpm` package is located in the **Packages** directory of your installed Linux iso file.

14. Change the files' access privileges. When the files were copied over in [step 4](#), their access privileges were changed to Read Only by default. Change the access privileges to Read/Write/Execute by calling the following command in the listed directories:

```
<purge_install>/linux/PurgeServices/bin> chmod 777 *
<purge_install>/lib> chmod 777 *
```

Configuring Purge Installations

If you have Purge installed, perform the following steps before accessing FTTPC Administrator.

1. Open the `AdmindedService.properties` file for Purge located at `<Purge_install>\<win or linux>\PurgeServices\conf`. If you installed Purge to the default location, the file is located at `C:\Rockwell\Purge\<win or linux>\PurgeServices\conf`.

2. Set the `FTPC_ADMIN_JMS_URL` property to match the `FTPC_ADMIN_JMS_URL` property value defined in FTPC Administrator's *ftpcAdmin.properties* file.

IMPORTANT: If all of your components (FTPC Administrator, Live Transfer, Purge, application server, etc.) are using the same ActiveMQ instance, the JMS URL for each component must point to the same ActiveMQ instance regardless of where the component is installed.

3. Define the component's service ID by setting the `SERVICE_ID` property. The ID for each component must be unique among all the Live Transfer, Loader, and Purge services.
4. Define the `SERVICE_ON` property.
 - ▶ Set this property to *true* if you want both the wrapper and the service to start automatically. When the service starts, it will look for any configured Purge jobs and their schedules. It will run the jobs as defined by their schedules.
 - ▶ Set this property to *false* if you want only the wrapper to start automatically. In this case, the service will not be started. Therefore, to start a Purge job, go to the Service Summary page for a list of configured Purge jobs and click [Run Job Now]. See “Starting and Stopping Purge Jobs” on page 184 for details.
5. Save and close the file.

Registering and Starting the Purge Service

NOTE: Please note that Purge and FTPC installed using the JBoss Standalone installer each install an instance of ActiveMQ that use the same port by default (61616). If both these installations are running on one machine, you only need to have one of the ActiveMQ instances running. Make sure both instances are not trying to run at the same time, which can result in a port conflict.

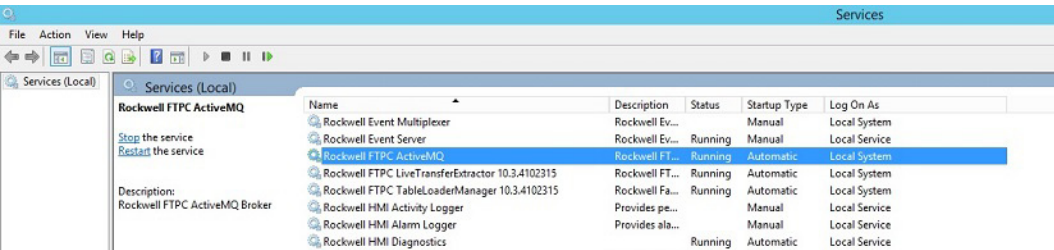
To register and start the Purge service on Windows or Linux, perform the steps listed in the following sections.

Windows

To register and start the Purge service on a Windows machine, perform the following steps in your Windows Start menu:

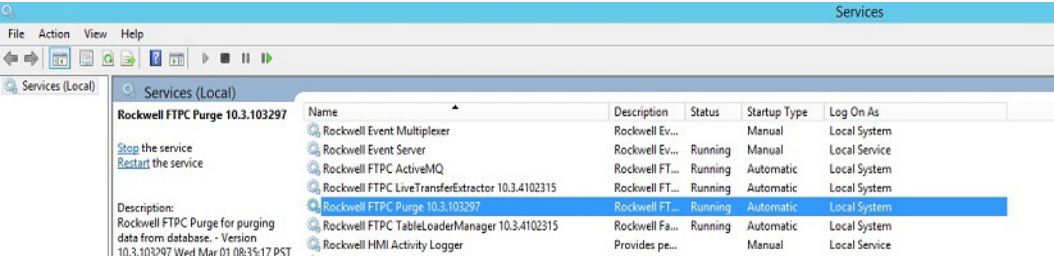
1. To register ActiveMQ as a service, go to Rockwell Software > FTPC Purge <version> > ActiveMQ > Register Service. If you are using both Live Transfer and Purge, you only have to register ActiveMQ once.
Once registered, the ActiveMQ service appears in the Services list.

Figure 5-7: ActiveMQ Service



2. Start ActiveMQ by going to Rockwell Software > FTPC Purge <version> > ActiveMQ > Start Service.
3. Register Purge as a service by going to Rockwell Software > FTPC Purge <version> > Purge Services > Register Service. Once registered, the Purge service appears in the Services list.

Figure 5-8: Purge Service



TIP: If you want to run a Purge job without registering Purge as a Windows service, go to FTPC Purge <version> > Purge Services > RunPurgeServices.

4. Go to Rockwell Software > FTPC Purge <version> > Purge Services > Start Service.

Linux

To register and start the Purge service on a Linux machine, perform the following steps:

1. If you are on Linux 7.0, update the wrapper files by performing the steps below. If not, skip to [step 2](#):
 - a. Go to following website to download the 64-bit version wrapper for Linux: <http://wrapper.tanukisoftware.com/doc/english/download.jsp> .
 - b. Go to the *bin* folder of the wrapper installer you downloaded in [step a](#) and copy the wrapper file.
 - c. Replace the wrapper file in *<Purge_install>/linux/PurgeServices/bin* by pasting in the new wrapper file.
 - d. Go to the *lib* folder of the wrapper installer you downloaded in [step a](#) and copy the wrapper.jar file.
 - e. Replace the wrapper.jar file in *<Purge_install>/lib* by pasting in the new wrapper.jar file.

2. To register ActiveMQ as a service that will automatically start on boot, run the following commands:

```
ln -s /rockwell/apache-activemq-<version>/bin/linux-x86-64/activemq
/etc/init.d/activemq
#chkconfig --add activemq
#chkconfig activemq on
```

If you are using both Live Transfer and Purge, you only have to register ActiveMQ once.

TIP: If you want to start ActiveMQ without registering it as a service, run the following command:

```
<Purge_install>/linux/JMS-Broker/apache-activemq-<version>/bin/activemq
```

3. Copy the runPurgeServices.sh file located at *<Purge_install>/linux/PurgeServices/bin/* to the *etc/init.d* directory and rename it to runPurgeServices. Then, run the following commands:

```
#cd /rockwell/Purge/linux/PurgeServices/bin
#cp -a runPurgeServices.sh /etc/init.d/runPurgeServices
```

4. Open runPurgeServices with a text editor and modify the APP_PATH parameter as shown:

```
APP_PATH="/rockwell/Purge/linux/PurgeServices"
```

5. To register Purge as a service that will automatically start on boot, run the following commands:

```
#chkconfig --add runPurgeServices
#chkconfig runPurgeServices on
```

TIP: If you want to run a Purge job without registering it as a service, run the following command:

```
<Purge_install>/linux/PurgeServices/bin/runPurgeServices.sh
start
```

Once Purge has started, the Purge jobs are run according to the schedules defined when the Purge job is created. If you want to run Purge immediately, go to Start > Programs > Rockwell Software > FTPC Purge <version> > Purge Services > RunJobNow. A command window opens and asks you to select the job that you want to run.

Stopping and Unregistering the Purge Service

Perform the steps in the following sections to stop and unregister the Purge service.

1. To stop Purge, perform the following:
 - ▶ **Windows:** go to Start > Programs > Rockwell Software > FTPC Purge <version> > Purge Services > Stop Service.
 - ▶ **Linux:** run the following command:


```
<Purge_install>/linux/PurgeServices/bin/RunPurgeServices.sh
stop
```
2. To unregister Purge, perform the following:
 - ▶ **Windows:** go to Start > Programs > Rockwell Software > FTPC Purge <version> > Purge Services > Unregister Service.
 - ▶ **Linux:** run the following command:


```
#chkconfig --del runPurgeServices
```
3. To stop ActiveMQ, perform the following:
 - ▶ **Windows:** Go to Start > Programs > Rockwell Software > FTPC Purge <version> > ActiveMQ > Stop Service.
 - ▶ **Linux:** Press [Ctrl] + C.
4. To unregister ActiveMQ, perform the following:
 - ▶ **Windows:** go to Start > Programs > Rockwell Software > FTPC Purge <version> > ActiveMQ > Unregister Service.
 - ▶ **Linux:** run the following command:


```
#chkconfig --del activemq
```

Viewing Purge Logs

TIP: In this section, *<Purge_install>* is where Purge is installed.

For errors related to Purge, look in the DsDataManagementServerLog*.html file located in the following folders:

- **Windows:** *<Purge_install>*\win\PurgeServices\logs\DataManagement
- **Linux:** *<Purge_install>*/linux/PurgeServices/logs/DataManagement

For errors related to the administering of Purge, look in the *<Purge_install>*\<os>\PurgeServices\logs\DataManagementAdminServer\dsDataManagementAdminServer*.html files, where *<os>* is either **win** or **linux** depending on your operating system.

Once a log file reaches a certain size, the logging information is sent to a new log file, so multiple files will be present in this directory. To locate the most recent log file, look at the file with the latest timestamp.

Purge information is stored in the database in the PURGE_LOG and PURGE_TABLE_LOG tables. Purge information is also stored in the ODS_REAPER_RECORD and ODS_MARKER_RECORD if Purge marking is done by the ODS marker.

Installing FTPC Administrator

In this chapter:

- ❑ **Installation Prerequisites 92**
 - Installing and Configuring ActiveMQ 92
 - Installing and Configuring Tomcat 92
 - Undeploying the Previous Application 93
- ❑ **Installing FTPC Administrator 93**
 - Installing on JBoss Standalone 93
 - Installing on JBoss Advanced or WebSphere 93
- ❑ **Configuring Data Management Installations 94**
- ❑ **Accessing FTPC Administrator 95**
- ❑ **Changing Passwords 97**

Installation Prerequisites

Before installing FTPC Administrator, you must install and configure Apache ActiveMQ and Tomcat. Please see the *FTPC Supported Platforms Guide* for supported versions.

If you have installed the JBoss Standalone version of FTPC, note the following:

- ☐ These applications are installed for you.
- ☐ The configuration files are located in the **activemq** and **tomcat** folders in `C:\Rockwell\PO<version>.<build>\` by default.
- ☐ ActiveMQ and Tomcat are started automatically when you start the application server.

Installing and Configuring ActiveMQ

Install ActiveMQ on the client machine that will be running FTPC Administrator. For more details, please refer to the official Apache ActiveMQ documentation.

IMPORTANT: ActiveMQ must be running in order to launch FTPC Administrator.

Installing and Configuring Tomcat

1. Install Tomcat on the client machine that will be running FTPC Administrator. See your Tomcat documentation for details.
2. Create an environment variable on your system as follows:

```
CATALINA_HOME=<Tomcat_install>
```

This environment variable can be either a User or System variable.

3. To avoid the serialization error messages when starting or stopping the Tomcat instance that is hosting FTPC Administrator, disable session persistence by opening the `context.xml` file located in `<Tomcat_install>\conf`.
4. Uncomment the following line:

```
<Manager pathname="" />
```
5. Save and close the file.
6. Start Tomcat by double-clicking the `startup.bat` file located at `<Tomcat_install>\bin`.

IMPORTANT: Tomcat must be running in order to launch FTPC Administrator.

Undeploying the Previous Application

If you have a previous version of FTPC Administrator installed and you are installing on JBoss Advanced or WebSphere, undeploy the previous application before installing the new one.

1. Open the following page in a browser:
`http://<tomcat_system>:<port_number>`
where `<tomcat_system>` is the name of the computer where you installed Tomcat and `<port_number>` is the port on which you configured Tomcat to listen.
2. Click Manager App on the right side of the page.
3. If prompted, enter the username and password for the Tomcat administrator. The username and password can be found in the `tomcat-users.xml` file located at `<tomcat_install>\conf`.
4. Look for the `FTPCAdmin.war` file, and then click the Undeploy link in the Commands column.
5. Stop, then restart the Apache Tomcat service.

Installing FTPC Administrator

Installing FTPC Administrator varies depending on your application server platform.

Installing on JBoss Standalone

When you install FTPC using the JBoss Standalone installation, FTPC Administrator will be installed automatically for you at `C:\Rockwell\PO<version>.<build>\FTPCAdminHome` by default. Two folders, called **conf** and **db**, are created automatically to contain the configuration and database folders.

NOTE: Do not delete the **conf** and **db** folders.

When you have Live Transfer and Purge jobs configured, they will be placed in a folder called **adminData** in the same location as the **conf** and **db** folders.

Installing on JBoss Advanced or WebSphere

If you are installing FTPC using either the JBoss Advanced or WebSphere installation, perform the following steps to install FTPC Administrator.

1. If Tomcat is running, stop Tomcat.

2. Create an FTPC Administrator home directory. For example:
C:\ftpcadmin\home.
3. In the FTPC Administrator home directory, create two folders called **conf** and **db**. These folders will contain the configuration and database folders.

NOTE: Do not delete these folders once they are populated.

When you have Live Transfer and Purge jobs configured, they will be placed in a folder called **adminData** in the same location as the **conf** and **db** folders.

4. Browse to the location where you installed FTPC and copy the *FTPCAdmin.war* file. The default installation directory is C:\Program Files\Rockwell Software\FactoryTalk ProductionCentre.
5. Paste the *FTPCAdmin.war* file into <Tomcat_install>\webapps.
6. Start Tomcat. A folder called *FTPCAdmin* is created in the *webapps* folder.
7. Stop Tomcat.
8. Open the web.xml file located at
<Tomcat_install>\webapps\FTPCAdmin\WEB-INF.
9. Search for the following text:

```
<param-value>y:/PlantOperations/bldFTPCAdmin/code/home</param-value>
```
10. Change the value to your FTPC Administrator home. For example:

```
<param-value>C:\ftpcadmin\home</param-value>
```
11. Save and close the file. Your changes will persist when Tomcat is restarted.
12. Copy the *ftpcAdmin.properties* and *logging.properties* files from
<Tomcat_install>\webapps\FTPCAdmin\WEB-INF\lib to
<FTPCAdmin_home>\conf.
13. Open the *ftpcAdmin.properties* file that you just copied into the conf folder.
14. Set the FTPC_ADMIN_JMS_URL value to your environment's hostname and ActiveMQ port number. For example:

```
tcp\://ussjcTestMachine\:61616
```
15. Restart Tomcat.

Configuring Data Management Installations

If you have Live Transfer and Purge installed, make sure you have performed the steps in the following sections before accessing FTPC Administrator:

- ☐ “Configuring Live Transfer Installations” on page 71
- ☐ “Configuring Purge Installations” on page 84

IMPORTANT: If all of your components (FTPC Administrator, Live Transfer, Purge, application server, etc.) are using the same ActiveMQ instance, the JMS URL for each component must point to the same ActiveMQ instance regardless of where the component is installed.

Before starting FTPC Administrator, make sure that your Live Transfer and Purge components are running. See “[Registering and Starting Live Transfer Components](#)” on page 73 and “[Registering and Starting the Purge Service](#)” on page 85 for details.

Accessing FTPC Administrator

IMPORTANT: Before accessing FTPC Administrator, make sure ActiveMQ and Tomcat are running. If you have Live Transfer and Purge installed, they should also be running before accessing FTPC Administrator.

To access FTPC Administrator:

1. Browse to the following URL:

`http://<hostname>:<port>/FTPCAdmin/FTPCAdmin`

where:

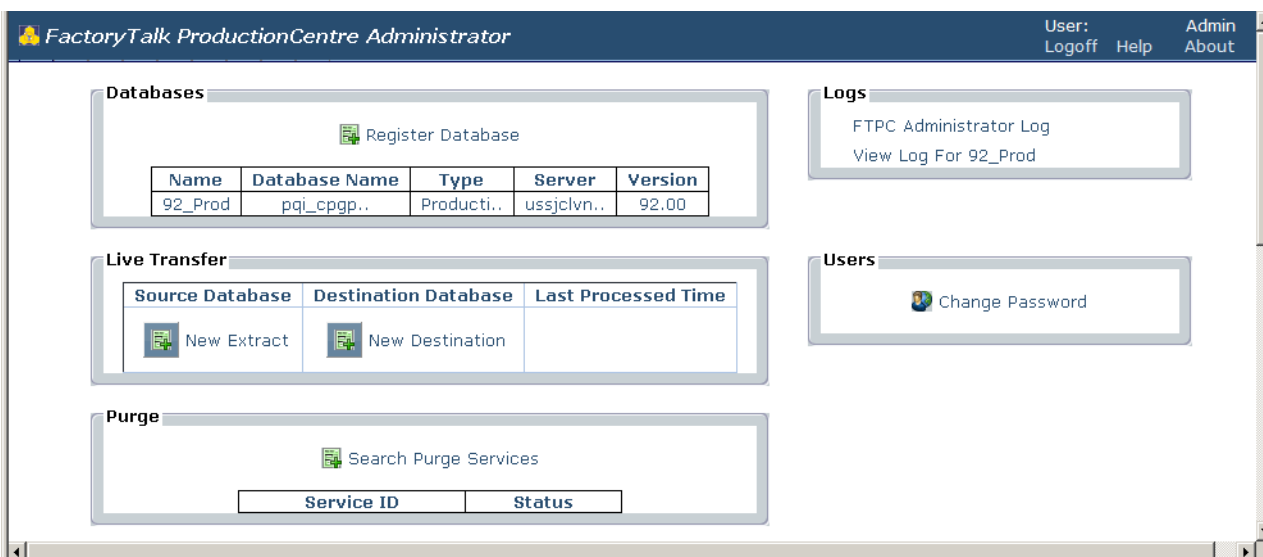
- ▶ *<hostname>*: the host name of the machine where you have installed FTPC Administrator.
- ▶ *<port>*: the Tomcat port as defined in the *server.xml* file located in *<Tomcat_install>\conf*. The default value is 8081.

TIP: If you have installed FTPC using either the JBoss Advanced or WebSphere installer, you can configure the URL to FTPC Administrator by configuring the **ftpcAdminURL** property in the *productioncentre.properties* file located where you installed FTPC (C:\Program Files\Rockwell Software\FactoryTalk ProductionCentre by default).

Figure 6-1: FTPC Administrator Login

2. Enter *admin/admin* for the username and password and click [OK]. Currently, this is the only user that can access the FTPC Administrator. You cannot create a new user to access this URL.

After you log in, the FTPC Administrator home page displays.

Figure 6-2: FTPC Administrator Home Page

From here, go “Registering a Database” on page 101 to continue your configuration.

To access help documentation, click the *Help* link in the upper right-hand corner.

IMPORTANT: If you restart ActiveMQ, you must restart the Tomcat service and re-log into FTPC Administrator after ActiveMQ has been restarted.

Changing Passwords

Upon installation of FTPC Administrator, a default administration user and password (admin/admin) are created that allows you to log into FTPC Administrator. You can change the password of this user in FTPC Administrator, but you cannot change the username.

To change the administration user's password:

1. On the FTPC Administration home page, click **Change Password**. The Change Password screen displays.

Figure 6-3: Change Password Screen

The screenshot shows a web form titled "Change User Password". It contains the following elements:

- A header bar with the title "Change User Password".
- A message: "You are about to Change the Password for User: admin".
- A section titled "User Password" containing three input fields:
 - "Original Password:"
 - "New Password:"
 - "Confirm New Password:"
- A "Change Comment:" label next to a text area.
- At the bottom right, there are two buttons: "OK" and "Cancel".

2. Enter the original password, new password, and confirmed password, and click [OK]. A dialog displays indicating that the password was successfully changed.

Working with Databases

In this chapter

- ❑ **Production Database** 100
- ❑ **ODS Database** 100
- ❑ **Registering a Database** 101
 - Editing a Registration 105
 - Deleting a Registration 105
- ❑ **Initializing the Database** 106
- ❑ **Migrating a Database** 107
 - Migrating a Base ODS 108
- ❑ **Updating a Database** 109
- ❑ **Reorganizing/Remapping a Database** 110
 - Understanding Remapping 110
 - Understanding Reorganizing 111
 - Performing the Remapping/Reorganization 112

Before you can get started using the FTPC applications (Process Designer, Shop Operations, etc.), you must have your databases set up. In order to run Live Transfer, you need to set up at least two databases [a Production database and an Operational Data Store (ODS) database]. When setting up your databases initially, you must:

- ☐ connect them to the application server. See the *Plant Operations Server Installation Guide* for your application server for instructions on creating the xml files that connect your database to FTPC.

IMPORTANT: Registering your databases in FTPC Administrator does not create the connection to the application server.

- ☐ initialize them.
- ☐ configure their site information.

Production Database

The Production database is a transactional processing database that records and stores all data collected from the Plant Operations product line and is the first location where all data is stored. The Production database cannot be the destination database in Live Transfer.

Periodically transferring data from the Production database to the ODS database (Live Transfer) and removing out-dated records from the Production database (Purge) maintains the performance of the Production database.

Multiple Production databases can share the same ODS database.

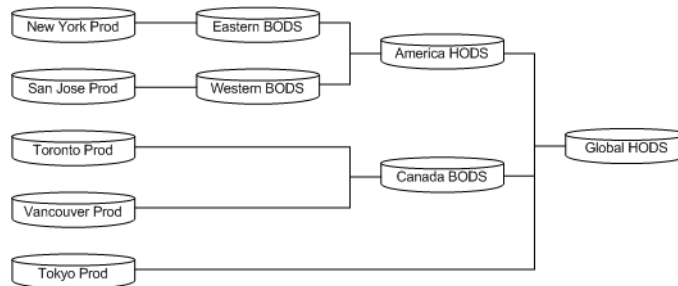
ODS Database

The ODS database (also known as the historical database) is the data warehouse where all data collected by Production databases is transferred during Live Transfer. It provides long-term storage and broad analysis of production information. You must have an ODS in order to run Live Transfer. Plant Operations supports the following types of ODSs:

- ☐ Base ODS (BODS): A base ODS consumes data from one or more Production databases.
- ☐ Hub ODS (HODS): A hub ODS consumes data from one or more Production databases, base ODSs, and hub ODSs.

The following figure shows a possible deployment of Production databases, BODs, and HODs.

Figure 7-1: Sample Database Deployment



Registering a Database

Before you can start configuring a database, the database must be registered in FTPC Administrator.

IMPORTANT: Registering a restored copy of a database that is currently registered within the same instance of FTPC Administrator is not allowed as they would both have the same GUID even if the two databases have different names. Two databases with the same GUID are not supported because Live Transfer and Purge depend on having a 1-to-1 database connection with FTPC Administrator via the GUID. To register a restored copy of a database, register the copy using a different instance of FTPC Administrator. The restored copy should also have its own ActiveMQ instance to handle messaging for its FTPC Administrator, Purge, and Live Transfer needs.

1. Click the **Register Database** link on the home page to go to the Database Registration screen.

Figure 7-2: Register Database Screen (SQL)

FactoryTalk ProductionCentre Administrator User: Logoff Help Admin About

Home
Initialize Database
Edit Configuration
Migrate Database
Reorganize Database
Update System Messages
Delete Runtime Data
Delete Work Order
Edit Registration
Delete Registration
View Consolidated Log

Database Registration

Name: 92_Prod
Description:
Type: Production
Vendor: MS SQL
Driver Type: type4
Server: ussjclvnguyen3
Database Name: pqj_cpjprod
User Name: sa
Password:
Port: 1433

Change Comment:

OK Cancel Verify

2. Enter the following information:

- ▶ **Name:** enter a name or ID for the administered component. The name should be unique among all administered components managed by this FTPC Administrator.
- ▶ **Description:** enter a description for the administered component for ease of identification.
- ▶ **Type:** select either *Production* or *ODS*.
- ▶ **Vendor:** select either *MS SQL* or *Oracle*.
- ▶ **Drive Type:** specifies the driver type to use for this connection.
 - ❖ If you selected MS SQL as your vendor type, select from the following options:
 - If this is a Microsoft SQL Server database, the driver type is *type4*.
 - If you want to set up database mirroring for a Microsoft SQL Server database, the driver type is *mirroring*.
 - If you want to set up AlwaysOn Availability Groups for a High Availability configuration, the driver type is *alwaysOnGroup*.

NOTE: AlwaysOn Availability Groups for High Availability is not supported on SQL Server 2008.

- ❖ If you selected Oracle as your vendor type, select from the following options:
 - If this is an Oracle database, the driver type is *thin*.
 - If this is an Oracle database in a Real Application Clusters (RAC) configuration, the driver type is *thin(Oracle-RAC)*.

The following properties displayed depend on the Vendor and Driver Type chosen:

► **SQL**

- ❖ *(Type4 Only) Server:* enter the server name for the database.
- ❖ *(Mirroring Only) Principal and Mirror Server:* for both the principal server and the mirror server, enter the host node definition separated by a comma. For example, serverA,serverB. In this instance, serverA is the principal server, and serverB is the mirror server.
- ❖ *(AlwaysOn Groups Only) Listener Server:* enter the listener server's IP address.
- ❖ *Database Name:* enter the name of the database.
- ❖ *User Name:* enter the user name for a user who has privileges on this database.
- ❖ *Password:* enter the password that corresponds to the user name.
- ❖ *Port:* enter the port on which you will connect to the database. The default is 1433.

► **Oracle**

- ❖ *Server:* enter the server name for the database.
- ❖ *SID:* enter the system identifier (SID) for the database.
- ❖ *User Name:* enter the user name for a user who has privileges on this database.
- ❖ *Password:* enter the password that corresponds to the user name.
- ❖ *Port:* enter the port on which you will connect to the database. The default is 1521.

► **Oracle RAC**

- ❖ *Host:Port List:* enter the host node definition:port number separated by a comma for each member of the RAC. For example, host1:1521,host2:1521.
- ❖ *Service Name:* enter the service name for the database.

- ❖ *User Name*: enter the user name for a user who has privileges on this database.
- ❖ *Password*: enter the password corresponding to the user name.
- ❖ *ONS Node:Port List*: enter the ONS node definition:port number separated by a comma. For example, racnode1:4200,racnode2:4200.

► **Change Comment**: Enter a comment for this action.

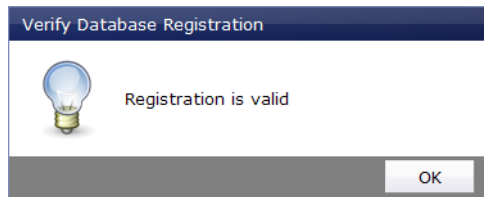
All fields are required except for the **Description** and **Change Comment** fields.

3. Click [Verify] to test the database connection.

NOTE: You must verify the database connection in order to save your registration. The OK button will not be enabled until the database connection is verified.

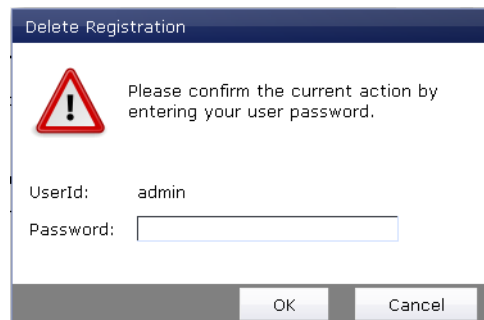
If the Administered Component (Service) is not running, the verification will fail even if the information is correct. A dialog will be displayed to inform you if the test succeeded or failed.

Figure 7-3: Valid Database Registration

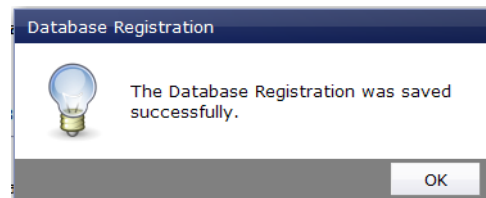


4. Click [OK] to close the registration dialog and [OK] again to save the registration. A dialog asking for your password displays.

Figure 7-4: Enter Password to Save Changes



5. Enter your password and click [OK]. A registration confirmation dialog displays.

Figure 7-5: Registration Confirmation Dialog

6. Click [OK] to close the confirmation dialog.

After the database has been registered, it will be displayed on the home page. Clicking on the link for the registered database in the Databases box on the home page will navigate to a manager interface where features can be configured for the selected database.

Editing a Registration

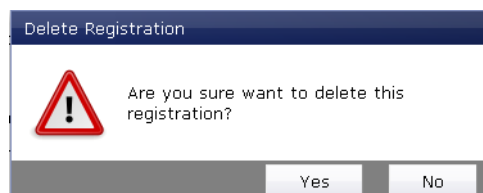
To edit a registration:

1. Select the database on the home page and then click **Edit Registration**. The database's registration configuration displays.
2. Make your changes and click [Verify]. A dialog displays indicating if the verification succeeded or not.
3. Click [OK] to close the verification dialog and click [OK] to save your changes. You will be asked to enter your password.
4. Enter your password to update the registration. A dialog displays indicating if the registration change was successful.
5. Click [OK] to close the dialog.

Deleting a Registration

To delete a registration:

1. Select the database on the home page and then click **Delete Registration**. You will be asked to confirm this deletion.

Figure 7-6: Deletion Confirmation

2. Enter your password to delete the registration.

Initializing the Database

If you are working with a new database, you must initialize it. The other database management options (migrating, reorganizing, etc.) are not available if your database is not initialized. If you select a database on the home page that has a version of SCHEMA_NOT_PRESENT, then you will automatically be taken to the Initialize Database screen.

When you initialize a database, all data (both runtime and buildtime) is deleted and all the tables that FTPC needs to store the data will be created. Always back up the database before initializing it.

IMPORTANT: Initializing the Production database resets values used by the Live Transfer process. If you are using Live Transfer and you reinitialize your Production database, then you must either reinitialize your ODS or change the site number of your Production database.

You must initialize a Production database before you can use it to store data. To do this:

1. Select the database on the home page, and then click **Initialize Database**.
2. Configure the following:
 - ▶ In the **Table Space Mapping** section, assign database tables (Oracle only) or filegroups (MS SQL Server only) to each index and table group.
 - ▶ In the **Site Info** section, set the database's site number and site ID. This site number must be unique across all (Production and ODS) databases. Once a site number is set, it cannot be changed unless you re-initialize your database. The site ID can be changed after it is saved.
 - ▶ **Change Comment:** Enter a comment for the action.
3. Click [OK]. A confirmation dialog displays.
4. Enter your password into the confirmation dialog and click [OK] to start the initialization. Once you have started the initialization, the progress is tracked on a progress bar in a dialog. After initialization is complete, the dialog closes.
5. Repeat steps [step 1](#) through [step 4](#) for any additional Production and ODS databases that need to be initialized.
6. Stop and restart your application server and all running client applications. If you are running Tomcat, you must also restart Tomcat.

If initialization fails, following these steps:

1. Do one of the following:
 - ▶ If you are using SQL databases, delete and recreate the database.

- If you are using Oracle databases:
 - a. Delete and recreate the Oracle user.
 - b. Re-assign any tablespaces as needed.
- 2. Run initialization again.

Migrating a Database

Schemas are the default table formats that come with the product. Some software versions include a schema upgrade. If you select a database on the home page that has a schema version that is not at least a 10.4 schema version, then you will automatically be taken to the Migrate Database screen.

IMPORTANT: Migration is only supported when the major schema version of the database to be migrated is different than the target major schema version. Migration is not available if the database being migrated has the same major schema version as the target major schema version. For example, you can use the migration utility to migrate from 9.3 to 10.4. However, migration is not available if you are migrating your database from a 10.42 schema version to a 10.43 schema version. For instructions on updating to another minor schema version, see [“Updating a Database” on page 109](#).

To migrate your database to a new major schema version, run the database migration wizard from FTPC Administrator. Database migration is recorded as an audit trail record in the database.

Before migrating your database:

- ☐ Make sure that the system is not in use (for example, no Live Transfers or client applications are running).
- ☐ Back up your database. If the migration fails, you must restore your database with backup data before attempting the migration again. If you do not restore your database after a failed migration, you will not be allowed to try the migration again.
- ☐ Check the database and server listed on the migration page to verify that you are migrating the correct database.
- ☐ Ensure that the data in your ODS matches the data in your Production database. To do this, run Live Transfer as needed.

To migrate an Oracle database, the user specified in the datasource definition must have the following privileges for both the Production database and ODS:

- ☐ CREATE INDEX
- ☐ ALTER PROCEDURE
- ☐ CREATE TRIGGER

- ☐ CREATE TABLE
- ☐ CREATE PROCEDURE
- ☐ CREATE VIEW
- ☐ EXECUTE PROCEDURE
- ☐ UNLIMITED TABLESPACE

For more information regarding the user privileges required for migration, see the migration articles listed in DSSolutions. If you do not have access to DSSolutions, please contact Rockwell Automation customer support.

IMPORTANT: Before migrating your database, make sure that you have checked the tablespace mappings for your database and that all objects are valid. If the tablespace mapping is wrong, migration will fail.

To migrate a database:

1. Select the database on the home page, and then click **Migrate Database**.
2. Optionally, enter a change comment for this action.
3. Click [OK]. A confirmation dialog displays.
4. Enter your password into the confirmation dialog and click [OK] to start the migration. Once you have started the migration, the progress is tracked on a progress bar in a dialog. After migration is complete, the dialog closes.
5. Exit FTPC Administrator.
6. Stop and start the application server. If you are running Tomcat, you must also restart Tomcat.

Migrating a Base ODS

When migrating an existing pre-8.2 base ODS (BODS) to 10.4, the migration process creates a new column called *pd_xfr_update_pid* in any table that is loaded by Live Transfer. This column stores the original Production database's *xfr_update_pid* value. For the migrated BODS, this value is based on the information in the XFR_LOAD_LOG table. The migration script creates this column, but it does not populate the column. To initially populate the *pd_xfr_update_pid* column, you must configure and run the *populatePdXfrUpdatePid.bat* script after you have migrated your BODS using the steps covered above. This process is handled separately from the rest of the migration so that you can schedule the script to run when it will not interfere with other processes on your system. This file is located in *C:\Rockwell\PO<version>.<build>\plantOpsDBScripting.zip* by default.

NOTE: This file does not need to be run on a newly-initialized BODS.

To migrate an existing pre-8.2 BODS, perform the following steps.

1. Navigate to C:\Rockwell\PO<version>.<build>. Locate plantOpsDBScripting.zip and extract the file and its contents.
2. Open a command prompt and navigate to where you extracted the populatePdXfrUpdatePid.bat file.
3. Enter the following command:

```
populatePdXfrUpdatePid.bat historical <dbURL> <port>
```

where <dbURL> is one of the following:

- **SQL:** sql:type4:<host>:<database>:<user>:<password>
- **Oracle:** oracle:thin:<host>:<SID>:<user>:<password>

where:

- ❖ <host> is the name of the database server,
- ❖ <database>/<SID> is the name or SID of the ODS you will be populating, depending on your database type.
- ❖ <user> is the database user name.
- ❖ <password> is the database password.
- <port> is the database port number.

You can stop this command whenever you wish with Ctrl+C, and restart it by entering the command above. When the command is re-started, it will pick up where it left off. If the command is stopped for other reasons (power outage, system reboot, etc.), restart it by entering the command above.

Updating a Database

To update your database from one minor schema version to another minor schema version (for example, 93.02 to 93.03), run the database update wizard from FTFC Administrator. Database updates are recorded as an audit trail record in the database.

IMPORTANT: Updating a database is only supported when the major schema version of the database to be updated is the same as the target major schema version. Updating is not available if the database being updated has a different major schema version from the target major schema version. For example, you can use the update utility to update from schema version 10.42 to schema version 10.43. However, updating is not available if you want to update your database from 9.4 to 10.4. For instructions on migrating to another major schema version, see [“Migrating a Database” on page 107](#).

Before updating your database:

- ☐ Make sure that the system is not in use (for example, no Live Transfers or client applications are running).
- ☐ Check the database and server listed on the update page to verify that you are updating the correct database.

To update a database:

1. Select the database on the home page, and then click **Update Database**.
2. Optionally, enter a change comment for this action.
3. Click [OK]. A confirmation dialog displays.
4. Enter your password into the confirmation dialog and click [OK] to start the update. Once you have started the update, the progress is tracked on a progress bar in a dialog. After the update is complete, the dialog closes.
5. Exit FTPC Administrator.
6. Stop and start the application server. If you are running Tomcat, you must also restart Tomcat.

Reorganizing/Remapping a Database

Use FTPC Administrator to either reorganize or remap your database.

Understanding Remapping

If you are transferring data from your Production database to a new database that does not have the same tablespace mapping (for example, to a database that was upgraded to a newer platform that has a different tablespace mapping) and the tablespaces in your new database do not match with the tablespaces in your Production database, you will get an error because exporting data from a Production database and importing it into another database will include the tablespace mapping of the original Production database. This error will also occur if you are transferring data from the new database to your historical database and their tablespaces do not match.

For example, if the Production database has any new tables that were created during Live Transfer or when a DCS or hierarchy was created, then the new table cannot be assigned to the tablespace defined in the new database because the tablespaces that are being assigned do not exist in the new database.

After restoring data into a database that has tablespaces that do not match those of the original database, assign current (and existing) tablespace names using FTPC Administrator so that you can go ahead and create the new objects.

TIP: You will not encounter this problem when you update the Production database with information from another database. In that situation, you will be using a DSX file rather than a database export. A DSX file only imports objects, not tablespace mappings.

Remapping tablespaces will change the values in the tablespace map table. This affects the location of future DCS table creations and hierarchy table creations. However, this will not re-locate tables. To re-locate tables, reorganize your database.

Understanding Reorganizing

When your data tables get too big, you may want to move them to different disks in order to speed up your access. FTPC Administrator allows you to “reorganize” your database by placing schema objects into different filegroups (MS SQL Server) or tablespaces (Oracle).

IMPORTANT: Prior to reorganizing a database, make sure that the system is not in use (for example, no Purges or Live Transfers are running) and all users are logged off.

The reorganization script recreates all of the tables, constraints, and indexes in the database using the grouping that you select through the Reorganize interface and invoke the commands that tell the database to recalculate statistics immediately so that the statistics will be up to date for any queries made against the database.

When you reorganize your database, the UDA and audit tables follow their main tables to the same tablespace. For example, if the UNIT table is moved to the fast growing tablespace, then the UNIT_A, UDA_UNIT, and UDA_UNIT_A tables all move to the fast growing tablespace.

The Reorganize utility groups the selected tables according to the user configured assignments for Oracle tablespaces or MS SQL Server filegroups. The application server assigns the tables and defines the growth type.

Consider these limitations:

- ☐ Audit tables and their indexes will go with their primary table. See Configuring Database Logging for more information on audit and primary tables.
- ☐ User-defined indexes must be recreated after reorganization.
- ☐ Oracle tables that have BLOBs in them will not be moved by the reorganize script. This type of table includes FORM_BLOB, DC_BLOB, and INSTRUCTION_BLOB for instance.

Performing the Remapping/Reorganization

To reorganize or remap a database:

1. Set up the database. The setup depends on the database platform. The names you use in the setup populate the Table Space Mapping section of the configuration screen in FTPC Administrator.

- **Oracle:** We recommend that you create four tablespaces, two with large extent sizes and two with small extent sizes. The large extent size tablespaces should be assigned to fast growing tables and fast growing indexes. The small extent size tablespaces should be assigned to the slow growing tables and slow growing indexes. The tablespace name appears on the drop-down menu.

If the application collects a lot of data into data collection sets, then assign the data collection tables and indexes to the large extent tablespaces. Otherwise, put the data collection on the small extent tablespaces. For more information on tablespaces, see your database documentation.

NOTE: In order to reorganize a database on Oracle, the administrator must have and maintain the CREATE TABLE, CREATE TRIGGER, and UNLIMITED TABLESPACE privileges for both the production and historical databases. For more information regarding the user privileges required for reorganization, see the *Plant Operations Server Installation Guide* for your application server.

- **SQL:** Create two filegroups, one for indexes and one for data. These will appear automatically in the reorganization wizard. For more information on filegroups, see your database documentation.
2. In FTPC Admin, select the database to be remapped/reorganized on the home page and then click Reorganize Database.
 3. Define the following:
 - **Table Space Mapping:** remap/reorganize the database tables (Oracle only) or filegroups (MS SQL Server only) for each index and table group. The Reorganize utility groups the selected tables according to the user-configured assignments for Oracle tablespaces or MS SQL Server filegroups. The application server assigns the tables and defines the growth type.
 - ❖ **Data Collection Indexes:** Indexes for data collection tables are located on the tablespace/filegroup DSDCINDEX.
 - ❖ **Data Collection Tables:** DCS and Hierarchy runtime tables are located on the tablespace/filegroup DSDCTABLE. These tables vary by customer application.
 - ❖ **Fast- and Slow-Growing Indexes:** To generate an up-to-date list of fast- and slow-growing indexes in MS SQL Server, open Microsoft SQL

Server Management Studio, navigate to your database, and create a new query. Enter the following statement.

```
select object_name(sys.indexes.object_id) as TABLE_NAME,
sys.indexes.name as INDEX_NAME,
XFR_SCHEMA_OBJECT.index_tablespace as INDEX_TABLESPACE
from sys.indexes
join XFR_SCHEMA_OBJECT
  on sys.indexes.object_id = object_id( XFR_SCHEMA_OBJECT.object_name )
  and sys.indexes.type > 0
order by 3, 1, 2
```

If you are using an Oracle database, enter the following statement in SQL*Plus:

```
select user_indexes.table_name as TABLE_NAME, user_indexes.index_name as INDEX_NAME,
xfr_schema_object.index_tablespace as INDEX_TABLESPACE
from user_indexes
join xfr_schema_object
  on user_indexes.table_name = xfr_schema_object.object_name
order by 3, 1, 2
```

When your list is generated, note that the index name DSLARGEINDEX indicates a fast-growing index. DSSMALLINDEX indicates a slow-growing index.

- ❖ **Fast- and Slow-Growing Tables:** To generate an up-to-date list of fast- and slow-growing tables, enter the following statement in your database management software:

```
select object_name, data_tablespace
from XFR_SCHEMA_OBJECT
order by 2, 1
```

When your list is generated, note that the table name DSLARGETABLE indicates a fast-growing table. DSSMALLTABLE indicates a slow-growing table.

- ❖ **XFR Temporary Tables:** Not used with this release.
- **Reorganize Effect:** select which database tables you want this action to affect.
- ❖ If you want to remap your database, select *New Tables Only*.
- ❖ If you want to reorganize your database, select *New and Existing Tables*.

- **Change Comment:** enter a change comment for this action.
- 4. Click [OK]. A confirmation dialog displays.
- 5. Enter your password into the confirmation dialog and click [OK] to start the remapping/reorganization. Once you have started the action, the progress is tracked on a progress bar in a dialog. The dialog closes upon completion.
- 6. Stop and start the application server. If you are running Tomcat, you must also restart Tomcat.

Your reorganization will fail if:

- ☐ Your Production database and ODS schema versions do not match.
- ☐ You do not have enough allocated disk space.
- ☐ You have not created the filegroups in MS SQL Server or tablespaces in Oracle.

Configuring Database Logging

In this chapter

- ❑ **Object Revision Logging 116**
 - Configuring Database Logging 119
 - How Primary and Audit Tables Work Together 118
- ❑ **Transaction Logging 118**
- ❑ **History Logging 119**
- ❑ **Configuring Database Logging 119**
- ❑ **Configuring Auditing Options 121**
 - Enabling/Disabling Auditing for Specific Tables 121
 - Disabling Runtime Activity Set Auditing 123
- ❑ **Configuring Application Logging 124**
- ❑ **Viewing the Administrator Log 124**

To configure database logging:

1. Select a database on the home page.
2. Select **Edit Configuration** in the left panel.
3. In the Database Configuration screen, define the properties in the Database Logging section.

Figure 8-1: Database Logging

Database Logging	
Object Revisioning Level:	None
Transaction Logging Level:	Standard
History Logging Level:	Standard
Sublot Quantity History and Revisioning:	<input type="checkbox"/>
Consolidated Logging	<input checked="" type="checkbox"/>
Application Log Retention Period	30 days
Application Log Maximum Size	100,000 rows

4. Click [Save]. A dialog asking for your password displays.
5. Enter your password and click [OK].

IMPORTANT: Because more than one application server can be connected to one Production database, changing the logging configuration at one application server will affect all the other J2EE application servers. In order for configuration changes to take effect, all of the application servers connected to the Production database must be restarted after you have made your configuration changes.

Changes to the database logging configuration are recorded as an audit trail record in the database.

Users can configure the types of transactions to be logged from the interface on the client machine. Configuration changes do not need to be made at the application server level. When an administrator configures the transaction log to record a certain logging detail level, those logging level selections apply to the entire active site (the Production database and its associated application servers). Each logging configuration level can be specified independently of the other two logging configuration levels. There are no dependencies. For example, object revisioning can still occur even if transaction logging is disabled.

Object Revision Logging

When the object revisioning is enabled, the system maintains all of the table record revisions in the database and tags each revision with a transaction ID to identify which transaction caused the change. Only saved changes will be recorded by

object revisioning. The system maintains pairs of tables (a primary table and an audit table) for those tables that are affected by the object revisioning requirements. As a result, reporting applications can write queries that return the state of a database record at a particular point in time.

You can enable object revisioning at the following levels:

- ☐ **None:** Object revisioning is not enabled.
- ☐ **Forms, Subroutines, and Event Sheets:** The audit tables record an object revision each time a change to forms, subroutines, or event sheets is saved and checked in to Process Designer.
- ☐ **All Process Designer Objects:** The audit tables record an object revision when changes are saved and checked in for any object in Process Designer are saved.
- ☐ **All Objects:** The audit tables record a object revisions when changes are made to any application-accessible object and their named UDAs. This includes all runtime objects like runtime data collection sets.

Hierarchy objects and any table that does not have an audit table are not subject to object revisioning.

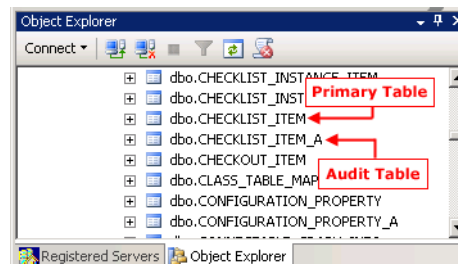
The database revision records created by the object revision preservation feature are transferred through Live Transfer.

Understanding Primary and Audit Tables

A primary table is a database table that is manipulated directly by the middleware and other components. It stores the current version of the objects. The `trx_id` column stores a transaction ID determined by the middleware. The value is associated with the transaction that caused the record in the primary table to either be inserted or updated. The value of the primary table's `trx_id` is always set for each transaction regardless of the configuration settings for application logging.

When object revisioning is enabled, a set of audit tables are available in the Production database that store old versions of an object after it has been changed.

Figure 8-2: Primary and Audit Tables



These audit tables are named as follows:

- ☐ `<tableName>` audit tables have the name `<tableName>_A`.

- ❑ DC_<objectName> audit tables have the name DA_<objectName>.
- ❑ UDA_<objectType> audit tables have the name UDA_<objectType>_A.
- ❑ AT_<applicationTableName> audit tables for the name AA_<applicationTableName>.

Audit tables are transferred to the historical database through Live Transfer and are also subject to the same purge criteria as their primary tables. See the Data Management Installation and Configuration Guide for more information on Live Transfer and Purge.

How Primary and Audit Tables Work Together

When object revisioning is enabled, creating new objects causes a new record to be written into the primary table but has no effect on the audit table. When updating or deleting an object, the original record is first copied into the audit table and then it is updated or deleted from the primary table(s). In addition to all of the primary tables' attributes, the audit tables have two additional attributes: audit_trx_id and audit_trx_time. When copying the record into the audit table, the value of the trx_id in the TRX_BASE table is set to the audit table's audit_trx_id identifying the transaction that is causing the record to be updated or deleted. The audit table's audit_trx_time is set to the time of the transaction that is causing the update or deletion.

This procedure of copying records to the audit table before they are changed in the primary table allows users to determine what changes were made to a record by comparing the records in the primary table to the records in the audit table where the primary table's trx_id is equal to the audit table's audit_trx_id.

For example, to find the state of a Unit record on Tuesday at 3:00 PM, the audit table can be searched for the record where the last_modified_time attribute is less than Tuesday at 3:00 PM and the audit_trx_time attribute is equal to or greater than Tuesday at 3:00 PM.

Transaction Logging

The transaction log can either be used to examine changes that were made during a particular period of interest or can be used to examine the detailed processing history of particular objects in the system. No errors or incomplete transactions are logged in the transaction log. This log can only be viewed in the database in the TRX_BASE table, not through FTFC Administrator.

Records in the transaction log cannot be modified or deleted. Each transaction and the steps associated with the transaction are tagged with a transaction ID, which is a GUID stored as a string value. The transaction log has the following additional attributes that identify the transaction:

- ❑ **Transaction ID (trx_id):** A GUID value that is associated with the changes made to data rows as a result invoking a method. For example, when calling a `Unit.finish()` method, all of the data modifications made as a part of this method transaction itself as well as any possible modifications and propagated transactions (`LotFinish`, `WorkOrderItemsFinish`, `WorkOrderFinish`) will be marked with the same transaction ID.
- ❑ **Method Name (method_name):** A string with the value “Class.method.” This string indicates the middle tier method that was invoked. For example, if `Unit.finish("finishReason")` was called, this attribute would log “Unit.finish.”
- ❑ **Method Parameters (method_params):** A string documenting the arguments passed to the API method. For example, if `Unit.finish("finishReason")` was called, the method parameters would be:
 - ▶ `trxTime = null`
 - ▶ `vUser = vUser`
 - ▶ `unitKey = 100`
 - ▶ `reason = finishReason`
 - ▶ `comment = null`
 - ▶ `validateRouteEnforcement = true`

History Logging

Administrators can choose to generate the `TRACKED_OBJECT_HISTORY` and `TOBJ_QUEUE_HISTORY` records or not. These are the database records that correspond to the `BaseHistory` and `QueueHistory` objects. History logging can be set to either “None” or “Standard.”

History logging is helpful when, for reporting purposes, you need to track how long a unit takes to get through each route step. The history logging feature matches up each start and complete transaction for each object at each step. This way, users can easily tell how long the unit stayed at each route step. If a unit is spending an abnormally long time at one step, the history logging feature will record this.

Configuring Database Logging

To set database logging:

1. In the **Database Configuration** screen, define the following application logging levels. These properties control how much information is recorded:
 - ▶ **Object Revisioning Level:** defines which objects will have revisions recorded to respective audit tables each time a change to the object is saved. The object revisioning level affects the check-out /check-in feature

in Process Designer. Object revisioning must be enabled for objects that you want to check-out and check-in. Once the appropriate level of object revisioning is enabled, the check out/check in feature will be enabled for those objects. See the Process Designer and Objects Help for more information on the check-out/ check-in feature.

The settings are:

- ❖ **None (0):** The audit tables do not record object revision information.
- ❖ **Forms, Subroutines, and Event Sheets (1):** The audit tables record an object revision each time a change to forms, subroutines, or event sheets is saved in Process Designer.
- ❖ **All Process Designer Objects (2):** The audit tables record an object revision when changes to any object in Process Designer are saved.
- ❖ **All Objects (3):** The audit tables record all object revisions when changes are made to any application-accessible object and its named UDAs. This includes all runtime objects like runtime DataCollectionSets.

TIP: Set the Object Revisioning Level to enable the check-in/check-out feature in Process Designer for the appropriate objects.

- ▶ **Transaction Logging Level:** defines which transactions are recorded in the TRX_ tables:
 - ❖ **Standard:** Applications log successful transactions, such as UnitStart and LotConsume.
 - ❖ **Standard + Test Data Collection:** Applications log successful transactions and data collection transactions, such as create, remove, and update.
- ▶ **History Logging Level:** defines whether information is logged to the transaction history tables:
 - ❖ **None:** Applications do not log information to the TRACKED_OBJECT_HISTORY and TOBJ_QUEUE_HISTORY tables.
 - ❖ **Standard:** Applications log information to the TRACKED_OBJECT_HISTORY and TOBJ_QUEUE_HISTORY tables.

2. Define subplot auditing and revisioning with the following properties:

- ▶ **Record Sublot History:** set this to *true* by clicking the checkbox if you want a SublotQuantityHistory object to be created when a subplot's quantity changes. This object tracks the quantity changes of the subplot. Use this object to understand the changes in a subplot's quantity over time and to provide the reason for the quantity change.

- ▶ **Enable Sublot Revisioning:** set this flag to *true* by clicking the checkbox if you want SublotRevision objects to be created for each subplot. These objects are audit copies of a subplot that allow you to retrieve specific instances of a subplot from the database.
3. Click [OK] to save your configuration.

Configuring Auditing Options

If you have object revisioning enabled, you can customize the auditing.

Enabling/Disabling Auditing for Specific Tables

You can enable and disable auditing on individual tables when the object revisioning level is set to any level except None. For example, if the level is set to *Forms, Subroutines, and Event Sheets*, you can disable auditing on the SUBROUTINE table. If the level is set to *All Objects*, you can disable auditing on specific objects by disabling auditing on their respective tables.

NOTE: Auditing for UDA tables is controlled by the master table. For example, if auditing is disabled on sublots (i.e., the SUBLOT table), then auditing is also disabled on the UDA_Sublot table.

To enable/disable auditing:

1. Set the object revisioning level. The level cannot be set to *None*. See “Configuring Database Logging” on page 119 for details.
2. Open your database management application.
3. In the XFR_AUDIT_OVERRIDE table, add entries that disable auditing for tables that you do not want audited. For example, if you want to disable auditing on the SUBLOT table, enter the following into the XFR_AUDIT_OVERRIDE table:
 - ▶ object_name: SUBLOT
 - ▶ object_type: Table (This value will always be Table.)
 - ▶ audit_type: 0

Figure 8-3: XFR_AUDIT_OVERRIDE Table

object_name	object_type	audit_type
SUBLOT	Table	0
NULL	NULL	NULL

For a list of the object revisioning levels that define the audit_type, see “Configuring Database Logging” on page 119.

4. Add entries that enable auditing for tables that are not currently audited. For example, the object revisioning level configured is Form, Subroutines, and Event Sheets, but you want a particular AT Definition table (AT_Example) to be audited. Enter the following into the XFR_AUDIT_OVERRIDE table:

- ▶ object_name: AT_Example
- ▶ object_type: Table
- ▶ audit_type: 1

For AT Definition tables, enter AT_<name> for the object name. For data collection tables, enter DC_<name> for the object name.

5. After all your entries have been made to the XFR_AUDIT_OVERRIDE table, run the dsResetAuditTriggers stored procedure against your selected database to reset the auditing triggers.

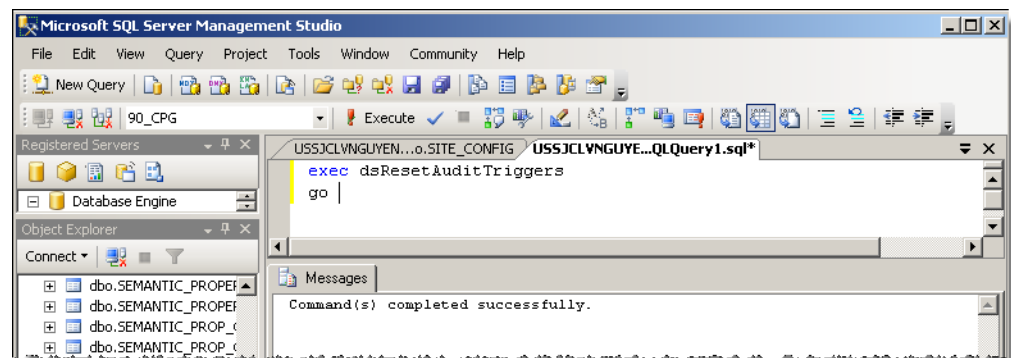
- ▶ If you are using a Microsoft SQL Server database, run the following:

```
exec dsResetAuditTriggers  
go
```

- ▶ If you are using an Oracle database, run the following:

```
declare  
begin  
    dsResetAuditTriggers () ;  
end;  
/
```

Figure 8-4: Resetting Audit Triggers



6. Restart your application server.

When importing a DSX file that will create or modify the runtime tables associated with AT Definitions, DCSs, STA Definitions, or UDA Definitions, the changes caused by the import will not automatically be reflected in the XFR_AUDIT_OVERRIDE table. To make sure that the changes are reflected in the table, use the following steps:

1. Insert the desired records into the XFR_AUDIT_OVERRIDE table. This should include any records that will affect the runtime tables that will be affected by the upcoming import (e.g., UDA_, STA_, AT_, DC_).
2. Start or restart your application server.
3. Import the DSX that will create or modify the runtime tables associated with AT Definitions, DCSs, STA Definitions, or UDA Definitions.

Performing these tasks in this recommended order allows the system to do the following:

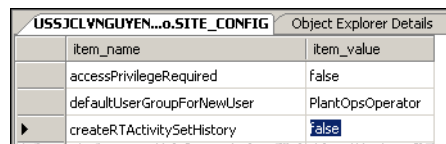
- ☐ When creating a runtime table, the middle-tier will use the XFR_AUDIT_OVERRIDE records to decide if an audit trigger should be created or not.
- ☐ When deleting or updating a record in the database, the middle-tier will use the XFR_AUDIT_OVERRIDE records to decide if a record should be inserted into the audit table or not.

Disabling Runtime Activity Set Auditing

You can disable runtime activity set records from being written to the RT_ACTIVITY_SET_HIS and TRX_BASE tables.

- ☐ **Disabling RT_ACTIVITY_SET_HIS Records:** History records are created in the RT_ACTIVITY_SET_HIS table for all runtime activity sets by default. To disable this feature and not write entries to the RT_ACTIVITY_SET_HIS table, set the createRTActivitySetHistory item in the SITE_CONFIG table to false. This must be done in the database table itself. Restart the application server for the changes to take effect.

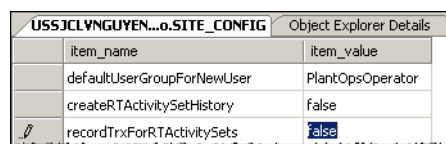
Figure 8-5: Disabling RT_ACTIVITY_SET_HIS Records



USSJCLVNGUYEN...o.SITE_CONFIG		Object Explorer Details
item_name	item_value	
accessPrivilegeRequired	false	
defaultUserGroupForNewUser	PlantOpsOperator	
createRTActivitySetHistory	false	

- ☐ **Disabling TRX_BASE Records:** If persistence is enabled on an activity set, a record is written to the TRX_BASE table each time a runtime activity set completes a step by default. If you do not want a record to be written to this table each time a step is completed, set the recordTrxForRTActivitySets item in the SITE_CONFIG table to false. This must be done in the database table itself. Restart the application server for the changes to take effect.

Figure 8-6: Disabling TRX_BASE Records



USSJCLVNGUYEN...o.SITE_CONFIG		Object Explorer Details
item_name	item_value	
defaultUserGroupForNewUser	PlantOpsOperator	
createRTActivitySetHistory	false	
recordTrxForRTActivitySets	false	

Configuring Application Logging

An Application Log stores an application's debugging messages associated with a Shop Operations Server client. Application Logs are located at the following locations depending on your environment:

- ☐ **JBoss StandAlone:**
C:\.FTPC*<serverName>*\ProductionCentre\logs\ApplicationLog
- ☐ **JBoss Advanced or WebSphere:** *<download_location>* \ProductionCentre\logs\ApplicationLog, where *<download_location>* is the location defined in the productioncentre.properties file. This is C:\.FTPC\AppServer by default..

The log for each client is only available on that specific client.

The messages logged by the Application Log are created in Process Designer using the Application object editor. You also have the option of enabling and disabling specific messages. Please see the *Process Designer and Objects Help* for more information on creating and enabling these application messages.

You can configure the Application Log by defining the following properties FTPC Administrator:

- ☐ **Application Log Retention Period:** indicates the number of days the Application Log messages will be saved before they are deleted. Select from 1 day, 7 days, 30 days (default), 6 months, and 1 year.
- ☐ **Application Log Maximum Size:** indicates the maximum number of rows that can be stored before the messages will be deleted. The default is 100,000 rows. Therefore, by default, only the latest 100,000 messages will be stored in descending order of insertion, and the rest are deleted. Select from 1,000 rows, 10,000 rows, 100,000 rows, and 1,000,000 rows.

Messages will be deleted if either one of these criteria are met. For example, there are only 500 messages, but these messages have been saved in the database for 31 days. All the messages will be deleted because one of the criteria has been met.

You can view this log in the Shop Operations Server Administration Console. See the *Plant Operations Server Installation Guide* for your application server for details.

Viewing the Administrator Log

Access the Administrator Log by clicking *FTPC Administrator Log* in the Logs section of the home page. This log displays the following:

- ☐ Login and logout actions
- ☐ Database initialization, migration, and reorganization actions
- ☐ Configuration changes

The log entries are listed chronologically from the earliest entry at the top to the most recent entry at the bottom. The log displays the following:

- ❑ **Date (YYYY/MM/DD HH:MM:SS):** the time the transaction was attempted.
- ❑ **Type:** Info, Error, or Warning.
- ❑ **Source:** the source of the transaction, such as Database Configuration Server.
- ❑ **User Name:** the name of the logged-in user that called the transaction.
- ❑ **Database Type:** the database type, either Production or ODS.
- ❑ **Site Number/ID:** the site number and site ID of the database.
- ❑ **Transaction:** the transaction that was performed.
- ❑ **Description:** a text record listing the details of the transaction.
- ❑ **Property/Old Value/New Value:** if a configuration value was changed, the property changed, the old value, and the new value.
- ❑ **Reason:** the reason for the transaction.

Use the buttons at the bottom of the log to scroll through the log pages. Click the refresh icon to refresh the log and return to the first page.

Figure 8-7: Administrator Log

Date (YYYY/MM/DD HH:MM:SS)	Type	Source	User Name	Database Type	Site Number/ID	Transaction	Description	Property/Old Value/New Value	Reason
2012/03/28 12:09:33 PM	INFO	Database Purge Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Purge Log	Check has installed Targeted Purge: sql:ussjclvnguyen3:1433:pqi_cpprod:sa		Check TP has been installed
2012/03/28 12:09:33 PM	INFO	Database Configuration Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Register Database	Check DB has been initialized: sql:ussjclvnguyen3:1433:pqi_cpprod:sa		Check DB has been initialized
2012/03/28 12:09:33 PM	INFO	Database Purge Server	admin	ODS	[185106035, USSJCPQI]	Purge Log	Check has installed Targeted Purge: sql:ussjclvnguyen3:1433:pqi_cpprod1:sa		Check TP has been installed
2012/03/28 12:09:33 PM	INFO	Database Configuration Server	admin	ODS	[185106035, USSJCPQI]	Register Database	Check DB has been initialized: sql:ussjclvnguyen3:1433:pqi_cpprod1:sa		Check DB has been initialized
2012/03/28 12:09:33 PM	INFO	Database Configuration Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Register Database	Check DB has been initialized: sql:ussjclvnguyen3:1433:pqi_cpprod:sa		Check DB has been initialized
2012/03/28 12:09:33 PM	INFO	Database Configuration Server	admin	ODS	[185106035, USSJCPQI]	Register Database	Check DB has been initialized: sql:ussjclvnguyen3:1433:pqi_cpprod1:sa		Check DB has been initialized
2012/03/28 12:09:22 PM	INFO	Database Configuration Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Save Production Database Configuration	Save database configuration for database at : sql:ussjclvnguyen3:1433:pqi_cpprod:sa	[Application Log Retention Period, 30, 7]	
2012/03/28 12:09:22 PM	INFO	Database Configuration Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Save Production Database Configuration	Save database configuration for database at : sql:ussjclvnguyen3:1433:pqi_cpprod:sa	[Merge Consumed Parts on Lot Merge, false, true]	
2012/03/28 12:09:22 PM	INFO	Database Configuration Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Save Production Database Configuration	Save database configuration for database at : sql:ussjclvnguyen3:1433:pqi_cpprod:sa	[Part Number Uniqueness in BillOfMaterials, true, false]	
2012/03/28 12:08:53 PM	INFO	Database Configuration Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Register Database	Check DB has been initialized: sql:ussjclvnguyen3:1433:pqi_cpprod:sa		Check DB has been initialized
2012/03/28 12:08:53 PM	INFO	Database Configuration Server	admin	ODS	[185106035, USSJCPQI]	Register Database	Check DB has been initialized: sql:ussjclvnguyen3:1433:pqi_cpprod1:sa		Check DB has been initialized
2012/03/28 12:08:53 PM	INFO	Database Purge Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Purge Log	Check has installed Targeted Purge: sql:ussjclvnguyen3:1433:pqi_cpprod:sa		Check TP has been installed
2012/03/28 12:08:53 PM	INFO	Database Configuration Server	admin	Production	[185107162, USSJCLVNGUYEN_92]	Register Database	Check DB has been initialized: sql:ussjclvnguyen3:1433:pqi_cpprod:sa		Check DB has been initialized

Configuring System Features

In this chapter

- ☐ **Updating the Site Number** 129
- ☐ **Setting Object Uniqueness** 130
- ☐ **Enabling/Disabling Consumed Parts Consolidation** 131
- ☐ **Enabling/Disabling Closing Consumed Lots** 131
- ☐ **Setting the Client Inactivity Timeout** 132
- ☐ **Configuring Access Control** 132
- ☐ **Configuring the PID Increment Interval** 133
- ☐ **Defining the Sublot Quantity Constraint** 134
- ☐ **Enabling/Disabling Client-Side Distributed Event Broadcasting** 134
- ☐ **Enabling/Disabling Server-Side Distributed Event Broadcasting** 136
- ☐ **Enabling/Disabling Keyed Object Changed Event Broadcasting** 138
- ☐ **Defining JMS Connection Information** 138
- ☐ **Configuring Messaging** 139
- ☐ **Configuring Filtering** 139
- ☐ **Enabling/Disabling SOAP Failover** 140
- ☐ **Configuring Sublot Naming in Inventory Batches** 140
- ☐ **Specifying Serial Numbers** 141
- ☐ **Setting up Email Services** 141
- ☐ **Modifying the User Login Requirements** 142
- ☐ **Updating System Messages** 143
- ☐ **Deleting Data** 143
 - Deleting Runtime Data 144
 - Deleting Work Orders 144

If you select a database on the home page that has been initialized and has a schema version of at least **93**, you are automatically taken to the View Configuration screen. To edit these options, click **Edit Configuration**.

You can configure the following features:

- ☐ “Updating the Site Number”
- ☐ “Setting Object Uniqueness”
- ☐ “Enabling/Disabling Consumed Parts Consolidation”
- ☐ “Enabling/Disabling Closing Consumed Lots”
- ☐ “Setting the Client Inactivity Timeout”
- ☐ “Configuring Access Control”
- ☐ “Configuring the PID Increment Interval”
- ☐ “Defining the Sublot Quantity Constraint”
- ☐ “Enabling/Disabling Client-Side Distributed Event Broadcasting”
- ☐ “Enabling/Disabling Server-Side Distributed Event Broadcasting”
- ☐ “Configuring Messaging”
- ☐ “Configuring Filtering”
- ☐ “Enabling/Disabling SOAP Failover”
- ☐ “Configuring Sublot Naming in Inventory Batches”
- ☐ “Specifying Serial Numbers”
- ☐ “Setting up Email Services”
- ☐ “Modifying the User Login Requirements”
- ☐ “Updating System Messages”
- ☐ “Deleting Data”

When you click [OK] to save the change, you will be asked to enter the administration password to save the change.

Any changes to these features are recorded as an audit trail record in the database.

Updating the Site Number

The site number and ID are defined when you initialize your database. The site number cannot be changed through FTPC Administrator, but you can change the site ID by going to the Site Info section. Enter the new site ID and click [OK].

Setting Object Uniqueness

To enable or disable part number uniqueness, go to the Runtime System section, and check or uncheck the **Part Number Uniqueness in BOM** property.

The part number uniqueness property specifies whether multiple revisions of the same part number are allowed in a BOM:

- ☐ If you clear the Part Number Uniqueness in the BOM checkbox, then users will be able to create a BOM containing multiple revisions of the same part number.

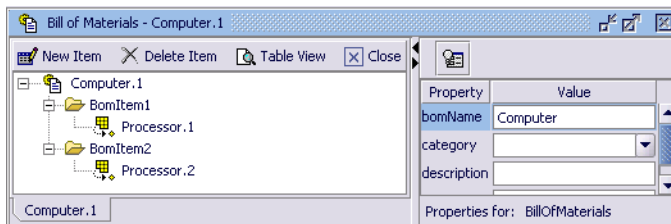
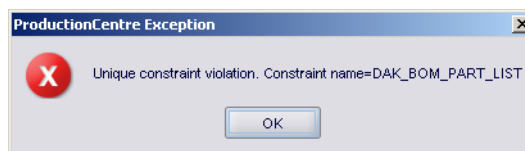


Figure 9-1: BOM without Object Uniqueness Defined

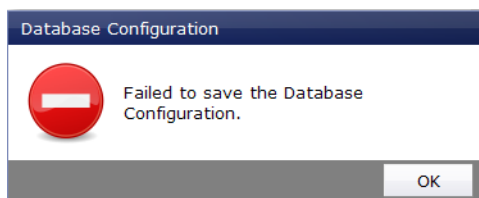
- ☐ If you check the Part Number Uniqueness in the BOM checkbox, then users will not be able to create a BOM object containing two revisions of the same part number. When you try to save a BOM with multiple versions of the same part, an error displays if the BOM contains two revisions of the same part number.

Figure 9-2: Object Uniqueness Error



If there are BOMs in your database that contain parts with identical numbers and you are trying to enable the constraint, you will not be allowed to enable this constraint. The following dialog displays:

Figure 9-3: Failed Configuration Change



The Part Number Uniqueness in a BOM constraint does not apply to process BOMs. Any changes to the object uniqueness configuration is recorded as an audit trail record in the database.

Enabling/Disabling Consumed Parts Consolidation

When merging lots, you have the option of consolidating the matching consumed parts as well. By default, the consumed parts are not merged when the lots are merged. Any changes to the consumed parts consolidation configuration is recorded as an audit trail record in the database.

To enable or disable this feature, go to the Runtime System section, and check or uncheck the **Merge Consumed Parts When Merging Lots** property.

When this feature is enabled, if each of the donor and target lots have a consumed part for a matching BOM item, then an attempt is made to consolidate them into a single consumed part. The following attributes determines whether consumed parts are the same:

- ☐ partNumber
- ☐ partRevision
- ☐ partSerial
- ☐ partBatch
- ☐ partInvoice
- ☐ placement

When this feature is disabled, if each of the donor and target lots have a consumed part for a matching BOM item, then the middleware will NOT attempt to consolidated them into a single consumed part. The number of consumed parts for the final merged lot will be the sum of those coming from the donor and target lots. The feature is disabled by default.

Enabling/Disabling Closing Consumed Lots

You can choose whether or not a subassembly lot closes automatically when its quantity_consumed value equals its quantity value after consumption.

To enable or disable this feature, go to the Runtime System section, and check or uncheck the **Close Lot When Completely Consumed** property.

When this feature is enabled, the subassembly lot's status/state automatically changes to Closed/Consume when the quantity_consumed value equals the quantity value.

When this feature is disabled, the subassembly lot's status/state does not change when the quantity_consumed value equals the quantity value.

IMPORTANT: If you are using the `BomItem.addConsumedPartForLot(lot, quantity, partInvoice, placement)` method to consume partial lots, then the subassembly partial lot's status/state automatically changes to Closed/Consume when the `quantity_consumed` value equals the `quantity` value regardless of the `Close Lot When Completely Consumed` setting.

Setting the Client Inactivity Timeout

You can set the Shop Operations client to log out the user after a certain interval of inactivity by setting the Client Inactivity Timeout property.

To enable or disable this feature, go to the Runtime System section, and define the timeout value in the **Client Timeout (Minutes)** property. The client timeout is disabled (i.e., set to zero) by default. The maximum value for the timeout is 60 minutes.

This timeout applies to all Shop Operations clients only. It does not apply to Process Designer or FTPC Administrator. You cannot set individual timeout intervals for specific runtime clients. However, you can disable the client inactivity timeout on individual clients. The client timeout interval applies when you are viewing forms through Shop Operations. It does not apply to forms viewed through form test mode in Process Designer.

The timer resets when any event is called, regardless of whether or not there is script behind the event. For example, if you click a form, the timer resets because the form's Click event was called.

Once the configured time interval is reached, the client times out and:

- ☐ any unsaved data in the forms is lost.
- ☐ all the forms in the form stack are closed.
- ☐ all the users in the user stack are logged out. If the user is part of a crew, the crew must be recreated.

After the client times out, two entries are logged in the `TRX_BASE` table: one for the client-side timeout and one for the actual logout transaction.

Configuring Access Control

Configure access control by going to the Runtime System section and defining the following properties:

- ☐ **Default User Group for New Users:** The user group assigned to a user object when it is first created if the administrator or user creation code does not

explicitly set the user group. By default, the following user groups are available with the following privileges:

- ▶ **PlantOpsGuest:** no privileges.
- ▶ **PlantOpsOperator:** user can perform operator runtime transactions. This is the default.
- ▶ **PlantOpsSupervisor:** user has all the PlantOpsOperator privileges and well as can change stations, disable auto timeout and time scripts, and open/close a form in Shop Operations.
- ▶ **PlantOpsDesigner:** user has all the PlantOpsSupervisor privileges as well as the ability to log into Process Designer and create, update, and delete Process Designer objects.
- ▶ **PlantOpsAdmin:** user has all the PlantOpsDesigner privileges as well as the ability to log into FTPC Administrator.
- ❑ **Explicit Access Privilege Required:** This checkbox applies to all runtime clients (e.g., Shop Operations or any Web Services interface). Check this box if you want to force a client to provide an explicit access privilege.
- ❑ **Authorization Cache Timeout:** This property defines how often the middleware checks to see if any user credentials have changed (for example, a user password has changed or the expiration date has passed). The default is 30 minutes. When using access control, changes to a user's credentials only take effect after the middleware has been notified of the change during one of these regular checks. For example:
 - ▶ The middleware has checked for changes to user credentials at 3:00.
 - ▶ The password for user “designer” has been changed at 3:05.
 - ▶ When logging into Shop Operations at 3:15, user *designer* must use their old password. The new password will not take effect until 3:30 when the next middleware check has occurred.
 - ▶ When logging into Shop Operations at 3:30, user *designer* can use their new password.

Configuring the PID Increment Interval

When applications create or update data, they store a process identifier (PID) with the data object. This allows the Live Transfer and Purge processes to handle the data in smaller sets and reduces extended periods of heavy load on the databases.

Configure the PID Increment Interval value depending on the amount of data you expect will be accumulated during the interval. For example, if you are collecting large amounts of data, you should adjust the PID increment interval to a smaller interval. This feature is intended to be used in situations where the transaction volume is large, which results in too much data being marked with the same PID.

You can configure the PID to increment more frequently so that less data is marked with a single PID value.

This configuration affects all datasources configured in that FTPC Administrator instance. When you configure it, these datasources will have their PIDs updated using the same configuration.

To configure the PDI increment interval, go to the Runtime System section and define the PID Increment Interval. Select from the following:

- ☐ 30 sec.
- ☐ 2 min.
- ☐ 5 min.
- ☐ 30 min.
- ☐ 1 hour (the default)

Defining the Sublot Quantity Constraint

Configure whether or not sublots can have a negative quantity by going to the Runtime System section and defining the **Sublot Quantity Constraint** property.

Select one of the following:

- ☐ **Require Positive Quantity:** choose this option if you want sublots to only have a positive quantity. This is the default.
- ☐ **Allow Negative Quantity:** choose this option if you want to allow sublots to have negative and zero quantities. In addition, this option also allows negative and zero quantities to be transferred between sublots.

NOTE: If you will be processing inventory batch transactions out of order, this property must be set to **Allow Negative Quantity** as there may be instances when an out-of-order transaction may result in a negative quantity. For example, if a ship transaction is called before a receive transaction is completed, a sublot's quantity might be a negative number before the receive transaction is completed.

Enabling/Disabling Client-Side Distributed Event Broadcasting

Change the **Enable Distributed Event** property to determine whether or not state change events are automatically broadcast to all registered listeners. Unlike the **Enable ServerStateChange Event** property, this feature is performed on the client side, which does not allow other applications (for example, Quality Management) to also listen for these state change events.

To subscribe for the state change event, register the `StateChangeEvent` listener using the `addStateChangeListener(StateChangeListener listener)` method. For example:

```
getServer().getFlexibleStateModelManager().addStateChangeListener(new StateChangeListener()
{
    public void onStateChange(StateChangeEvent event)
    {
        ObjectState currentState = event.getObjectState();
        label.setText(currentState.getState().getName());
        ...
    }
});
```

In addition, the `DistributedEventConfiguration` `DsList` object in Process Designer should be updated with the connection configuration to your chosen messaging service.

An object's state changes when one of the following transactions occurs:

- ☐ An object is created. If an automatic Flexible State Model (FSM) relationship is configured for the object type created, the object state is changed to the FSM's default state.
- ☐ The object's FSM changes.
- ☐ The `applyTransition(...)` method is called.
- ☐ The `changeState(...)` method is called.

NOTE: If any of these events does not result in the object changing states, no state change event is triggered.

By default, when a state change event occurs, it is not broadcast to all registered listeners. The object's state must be updated with a manual refresh. To update an object's state automatically when a state change event occurs, check the **Enable Distributed Event** property.

This configuration is stored in the **enableDistributedEvent** item in the `SITE_CONFIG` table. The possible values are *false* (default) or *true*.

IMPORTANT: If your messaging services uses ActiveMQ (see “[Configuring Messaging](#)” on page 139 for details), then you must provide the JMS connection information in the DistributedEventConfiguration DsList object in Process Designer. For example, CPG provides a sample JMS DsList object (JMS_ActiveMQ_Sample). The information in that object should be copied into the DistributedEventConfiguration DsList object if you choose to use ActiveMQ for messaging.

Enabling/Disabling Server-Side Distributed Event Broadcasting

Change the **Enable ServerStateChange Event** property to determine whether or not state change events are automatically broadcast to all registered listeners using an ActiveMQ topic. Unlike the **Enable Distributed Event** property, this feature is performed on the server side, allowing other applications (for example, Quality Management) to also listen for state change events.

To subscribe to a state change event,

1. Create a listener.

```
ServerEventListener listener = new ServerEventListener()
{
    @Override
    public void onServerEvent(FTPCEvent event)
    {
        //customer logic
    }

    public Class getEventClass()
    {
        return ServerStateChangeEvent.class;
    }
};
```

2. Create the JMS connection.

```
Connection connection = null;
ActiveMQConnectionFactory factory = new
```

```

ActiveMQConnectionFactory("tcp://10.85.106.55:61616");
try
{
    connection = factory.createConnection();
    connection.start();
}
catch (Exception e)
{
    e.printStackTrace();
}

```

3. Add the listener.

```

ServerDistributedEventManager.getInstance().addServerEventListener(listener);

```

4. Register the Java Eventing Distribution (JED).

```

ServerDistributedEventManager.getInstance().registerJavaEventingDistribution(connection);

```

An object's state changes when one of the following transactions occurs:

- ☐ An object is created. If an automatic Flexible State Model (FSM) relationship is configured for the object type created, the object state is changed to the FSM's default state.
- ☐ The object's FSM changes.
- ☐ The applyTransition(...) method is called.
- ☐ The changeState(...) method is called.

NOTE: If any of these events does not result in the object changing states, no state change event is triggered.

By default, when a state change event occurs, it is not broadcast to all registered listeners. The object's state must be updated with a manual refresh. To update an object's state automatically when a state change event occurs, check the **Enable ServerStateChange Event** property.

Because the server side distributed events use ActiveMQ, provide the JMS connection information. See “[Defining JMS Connection Information](#)” on page 138 for details.

This configuration is stored in the **enableServerDistributedStateChangeEvent** item in the SITE_CONFIG table. The possible values are *false* (default) or *true*.

IMPORTANT: If this feature is enabled, make sure that ActiveMQ is running.

Enabling/Disabling Keyed Object Changed Event Broadcasting

Change the **Enable Keyed Object Changed Event** property to determine whether or not the middle-tier will broadcast an event to all clients using the Distributed Event infrastructure each time a top-level buildtime object (i.e., any object that can be created and configured in Process Designer) has been saved or deleted. This property is enabled by default.

NOTE: When this feature is enabled, the cache of the web applications that runs on FTPC is automatically cleared when the object is updated.

Because the keyed object changed events use ActiveMQ, provide the JMS connection information. See “[Defining JMS Connection Information](#)” on page 138 for details.

This configuration is stored in the **enableKeyedObjectChangedEvent** item in the SITE_CONFIG table. The possible values are *false* or *true* (default).

IMPORTANT: If this feature is enabled, make sure that ActiveMQ is running.

Defining JMS Connection Information

NOTE: When defining the ActiveMQ resource adapter during your FTPC configuration, the configuration must match the JMS Server URL setting defined here.

For any feature that uses ActiveMQ, provide the JMS connection information by defining the following properties:

- ☐ **JMS UserName:** enter the user name for the JMS connection. If left blank, FTPC will use default values.
- ☐ **JMS Password:** enter the password for the JMS connection. If left blank, FTPC will use default values.

TIP: For ActiveMQ, the user name and password are optional.

- ☐ **JMS Server URL:** enter the server URL of the JMS connection. If ActiveMQ is not running on the local machine, change the default value of *localhost* to the hostname or IP address of the machine running ActiveMQ.

Configuring Messaging

FactoryTalk ProductionCentre provides two options for configuring messaging communication, JGroups messaging or ActiveMQ (JMS) messaging. JGroups and ActiveMQ messaging services will collect and process all messages that are sent within a unique site ID. The messaging service will only broadcast relevant messages to activities that share the same message topic name and DsList object.

IMPORTANT: If you change the messaging type to ActiveMQ, your existing JGroups configuration will be removed. Also, if you import objects from a previous build, your messaging configuration will be removed even if you are not changing your messaging type. Therefore, as a best practice, back up your existing configuration by exporting your existing messaging-configured objects into a DSX file before changing the messaging type or importing objects from a previous build.

To configure the messaging service:

1. Under the property `MessagingType`, select either `JGROUP` or `ActiveMQ`. By default, `JGROUP` is selected.
2. Click [OK]. A confirmation dialog displays.

IMPORTANT: Once you change your messaging service to ActiveMQ, any activities in your database that have the configuration items *messageConfigDsList* and *messageTopicName* will need to be reconfigured. Once you change your messaging service to JGROUP, any activities in your database that have the configuration items *jgroupClusterName* and *jgroupConfigDsList* will need to be reconfigured.

3. Enter your password into the confirmation dialog and click [OK] to change the messaging service.

Configuring Filtering

Define the **Max Rows in Filter** property to determine the maximum number of rows returned when filtering. This property is set to 0 by default, which means an unlimited number of rows can be returned.

NOTE: If the `Filter.setPagingFilterRowCount(...)` method is called, it takes precedence over the **Max Rows in Filter** configuration.

Enabling/Disabling SOAP Failover

NOTE: This section only applies to JBoss environments.

Use the **Allow SOAP Failover** property to enable/disable using the SOAP protocol in the following situations. This property is selected (i.e., enabled) by default.

- ☐ When logging into Shop Operations Server (SOS), if the RMI connection fails, the system will use the **Allow SOAP Failover** property configuration to determine if it should use SOAP. If enabled, the client fails over to SOAP. If disabled, the client will not fail over to SOAP.
- ☐ All user authentication and application transactions use SOAP by default. If you unselect the **Allow SOAP Failover** property, user authentication will still be done by SOAP, but all application transactions will use RMI. Unselect this property if your application uses features that are not supported by SOAP (for example, UserTransaction objects).

Configuring Sublot Naming in Inventory Batches

Change the **Specified Sublot Name in Inventory Batch** property to define how sublots in an inventory batch are named when the ProcessStep.produce(...) or ISublotContainer.receive(...) methods are called.

- ☐ When calling ProcessStep.produce(...), the produced sublot name is defined by the targetSublotName argument.
- ☐ When calling ISublotContainer.receive(...), the received sublot name is defined by the receivedSublotName argument.

Select one of the following:

- ☐ **Ignore for Joined:** use the sublot name defined by either the targetSublotName or receivedSublotName argument if the inventory batch has a management style of Layered, but ignore these arguments if the inventory batch has a management style of Joined. This is the default for a migrated database. When this value is selected, the produced/received sublot name is ignored if the target inventory batch is a Joined inventory batch. If the target inventory batch is a Layered inventory batch, a non-null value must be defined for the sublot name (using either the targetSublotName or receivedSublotName argument, respectively) and must be unique within the inventory batch.
- ☐ **Always:** always use the specified sublot name defined by the targetSublotName or receivedSublotName argument, respectively. This is the default for a newly-initialized database. When this value is selected, the produced/received sublot name is used if a non-null value is specified

regardless of the target inventory batch's management style. If a null value is specified, the system generates a subplot name regardless of management style.

This configuration is stored in the **inventoryBatchSublotNameSpecification** item in the SITE_CONFIG table. The possible values are 0 (Ignore for Joined) and 1 (Always).

Specifying Serial Numbers

Configure how units are assigned serial numbers by going to the Serial Number section and defining the following properties:

- ☐ **Series Counter:** The series counter shows the last serial number created by the application. If the value is 0, then the application has not created any serial numbers. Edit this field if you want to manually set the starting number for future serial numbers. This property is set to 0 by default.
- ☐ **Series Number of Digits:** The number of digits, including placeholders, to use in a serial number. For example, setting this property to 2 generates DS01, DS02, DS03, and so on. Setting this property to 3 generates DS001, DS002, DS003, and so on. This property is set to 0 by default.
- ☐ **Serial Number Prefix:** The prefix to appear before the digits in a serial number. For example, a prefix of RA generates RA001, RA002, RA003, and so on. This property is set to RA by default.

Setting up Email Services

The email service allows you to receive email for any results related to administrative procedures. You can also send email from within Process Designer or Shop Operations after enabling email services.

Configure the email service by going to the Email section and defining the following properties:

- ☐ **SMTP Host ID:** Enter the name of the SMTP server. The SMTP Server manages all the notification email messages sent regarding the core middle-tier. It manages the emails for the application server regardless of where a client machine is in the network. The application server contacts the SMTP server and sends the email.
- ☐ **Administrator E-Mail:** Enter the site administrator's email address. This is the email address that will receive all notifications.
- ☐ **Default From Address:** Enter the default address to send notifications from.
- ☐ **Maximum E-mail Attachment Size (MB):** Enter the maximum size that an email attachment can be. The default is 2 MB. Attachments can either be files or work instruction objects. If a file size is more than the maximum size set,

then an error will occur and the e-mail will not be sent. When setting the maximum email attachment size, take into account your system's hardware resources. Attaching a large file to your email may slow down your system.

To test the email service, click [Test Email]. This will automatically send an email from the address configured under the Default From Address property to the Administrator E-Mail Address property.

Modifying the User Login Requirements

If you do not have an existing security model in place, you can use the Plant Operations Custom Security Provider for security. If you are using the Plant Operations Custom Security Provider, you can configure the minimum password lengths and the number of login attempts that are allowed for each user. Changes to the security configuration are recorded as an audit trail record in the database.

Modify the user login requirements by going to the User Login section and defining the following properties:

- ☐ **Minimum Password Lengths:** To meet your security needs, you can define a minimum length for all application users. If a user tries to change their password to one that is shorter than the minimum length, then they will see an error message and must reenter a valid password. This property is set to No Limit by default.
- ☐ **Maximum Login Attempts:** You can define the number of failed login attempts a user is allowed before that user is locked out. Enabling this feature will record the number of failed login attempts by each user. When the count reaches the maximum, the user is locked out of Plant Operations. To remove the lockout, a user with administrator or designer privileges must reset the user status in Process Designer from 'Disabled' to 'Normal'. The lockout feature has a value of 0 so that it is disabled by default. This property is set to No Limit by default.

NOTE: If this feature is enabled, any action attempted by a user with an expired password will register as a failed login attempt.

- ☐ **Minimum Lowercase Character Length:** You can define the minimum number of lowercase letters required in a password. This property is set to No Limit by default.
- ☐ **Minimum Uppercase Character Length:** You can define the minimum number of uppercase letters required in a password. This property is set to No Limit by default.
- ☐ **Minimum Numeric Character Length:** You can define the minimum number of numeric characters required in a password. This property is set to No Limit by default.

- ❑ **Minimum Special Character Length:** You can define the minimum number of special letters required in a password. The special characters must be entered without using the Alt key (i.e., ~, ` , !, @, #, \$, %, ^, &, *, (,), _, -, +, =, {, }, [,], :, ;, ", ' , <, >, ., ,, ? /). This property is set to No Limit by default.

Updating System Messages

The PlantOperationsErrorMessage, PlantOperationsTrxMessage, and PlantOperationsUIMessage message object contains the IDs for errors returned in the response object. When you update your Plant Operations build, you can refresh this message object with changes that come with the upgrade without migrating your entire database. If you have provided messages in other languages for localization, these configurations will not be overwritten.

To update system messages:

1. Click Update System Messages.
2. Optionally, enter a comment for the action. This comment will be entered into the DS_ADMIN_LOG table as part of the audit control records.
3. Click [OK]. A confirmation dialog displays.
4. Enter your password into the confirmation dialog and click [OK] to start the update. Once you have started the update, the progress is tracked on a progress bar in a dialog. After update is complete, the dialog closes.

Deleting Data

You can select to delete all runtime data from the Production database or select to delete individual work orders. This allows you to quickly purge specific data or all runtime data from the database.

IMPORTANT: Deleting all runtime data through FTPC Administrator resets values used by Live Transfer. If you are using Live Transfer and delete all runtime data, then you must reinitialize your ODS. Contact Technical Support if you have any questions about this warning before you continue with this task. Deleting work orders through FTPC Administrator does not affect Live Transfer.

Deleting Runtime Data

IMPORTANT: When deleting all runtime data, make sure that you have no runtime data running. If you have runtime data running, an error will occur and not all the runtime data will be deleted.

Do not use FTPC Administrator to delete runtime data that has associated UDA references you want to keep. If you do, those UDA references will be lost.

To delete runtime data:

1. Click Delete Runtime Data.
2. Optionally, enter a comment for the action.
3. Click [OK]. A confirmation dialog displays.
4. Enter your password into the confirmation dialog and click [OK] to start the deletion. Once you have started the deletion, the progress is tracked on a progress bar in a dialog. After deletion is complete, the dialog closes.

Deleting Work Orders

To delete a specific work order:

1. Click Delete Work Order.
2. Enter the work order number in the **Work Order Number** text box.
3. Optionally, enter a comment for the action. This comment will be entered into the DS_ADMIN_LOG table as part of the audit control records.
4. Click [OK]. A confirmation dialog displays.
5. Enter your password into the confirmation dialog and click [OK] to start the deletion. Once you have started the deletion, the progress is tracked on a progress bar in a dialog. After deletion is complete, the dialog closes.

Configuring Live Transfer

In this chapter

- ❑ **Creating Live Transfer Instances 146**
 - Defining the Source Database 146
 - Defining the Destination Database 152
 - Creating Instances from Existing Configurations 153
 - Editing Live Transfer Configurations 154
- ❑ **Sending Live Transfer Configurations 154**
- ❑ **Using the Comparison Tools 156**
 - Using LTPIDCompare 156
 - Using LTSampleCompare 157
- ❑ **Starting and Stopping Live Transfer 158**
- ❑ **Configuring Live Transfer Logging 159**
- ❑ **Configuring ActiveMQ 160**
 - Configuring a Separate ActiveMQ Cluster 160
 - Configuring ActiveMQ for a Firewall Environment 160
- ❑ **Configuring ActiveMQ Monitoring 161**
- ❑ **Configuring Live Transfer with Linux 163**
 - Setting the Maximum Memory 164
- ❑ **Configuring the Java Service Wrapper (Optional) 164**
- ❑ **Deployment Options and Recommendations 165**
 - Extraction Recommendations 165
 - Loader Recommendations 166
 - ActiveMQ Recommendations 166
 - Deployment Scenarios 167

Use FTPC Administrator to configure the Live Transfer components (SEM, TEM, TLM). All databases that will be using Live Transfer must be registered in FTPC Administrator before they can be selected in a Live Transfer configuration. You must have at least two databases (one source and one destination) registered to configure Live Transfer. See “[Registering a Database](#)” on page 101 for details.

IMPORTANT: Once Live Transfer is started, changes to any of the configuration files will be ignored. To change any of the configurations, you must first stop Live Transfer, make the changes, and then restart Live Transfer. When Live Transfer is restarted, it will pick up where it left off before it was stopped as if there had been a power failure. See “[Handling a Communication Loss](#)” on page 197 for details.

Creating Live Transfer Instances

You can create Live Transfer instances starting with either the source database or the destination database.

Defining the Source Database

To define the source database:

1. In the Live Transfer box on the home page, do one of the following:
 - ▶ Click the **New Extract** icon if you are configuring the source database first.
 - ▶ Click the **Configure <Production_Name>** icon if you have already configured the source database.

The Live Transfer Extract screen displays.

Figure 10-1: Live Transfer Extract

2. Define the following properties:

- ▶ **Source Database:** select the database that will provide the information to be transferred. The drop-down menu is populated with all registered databases. The names listed are the names you defined when registering the database. They are not the actual database names.
- ▶ **Destination Database:** select the database that receive the information transferred. The drop-down menu is populated with all registered ODS databases. The names listed are the names you defined when registering the database. They are not the actual database names. If you defined the destination database first, it will be listed here. If you defined your destination database first, it will be selected here.
- ▶ **Live Transfer Extractor:** click the **Live Transfer Extractor** link to display a drop-down menu of available LTEs. Select an LTE from the list. This list is populated by the LTEs' service ID names as defined in their *AdmnedService.properties* file. See “Configuring Live Transfer Installations” on page 71 for details.
- ▶ **JMS Connection URL:** enter the connection information required to connect to the appropriate ActiveMQ broker in order to communicate with the TLM. The format should be as follows:

```
failover:(<hostname>:<port>)
```

For example: **failover:(tcp://mqhost:61616)** Please note that the hostname must include the appropriate protocol.

IMPORTANT: Using a failover URL format ensures that if a connection is dropped (for example, due to inactivity), reconnection will be attempted.

The default ActiveMQ port is 61616. Live Transfer and FTPC Administrator can use different ActiveMQ brokers.

Click [Test] to test the JMS connection. The connection must be running to perform the verification. However, the connection does not have to be running or verified to save the configuration.

- ▶ **JMS User:** enter the user name for a user who has privileges on the ActiveMQ broker. If a user name is not specified, the system will use ActiveMQ's default user name, which is *system*. Depending on your configuration, this property may be optional.
- ▶ **JMS Password:** enter the password corresponding to the user name defined in the **JMS User** property. If a password is not specified, the system will use ActiveMQ's default password, which is *manager*. Depending on your configuration, this property may be optional.
- ▶ **Transfer Type:** define the type of transfer.
 - ❖ **All:** all appropriate tables are transferred.
 - ❖ **List:** a specific list of tables is transferred. When this option is selected, the List Transfer screen displays:

Figure 10-2: List Transfer Screen

Live Transfer From Production (92Prod) to ODS (92ODS)

Tables	Tables to be transferred
SUBROUTINE	BATCH
SUBROUTINE_A	BATCH_A
SUBROUTINE_BLOB	SUBLOT
SUBROUTINE_BLOB_A	SUBLOT_A
TASK	SUBLOT_RELATION
TASK_GROUP	SUBLOT_RELATION_A
TASK_GROUP_TASK_REL	SUBLOT_STORAGE_HISTORY
TASK_ITEM	
TBOM	
TBOM_A	
TBOM_ALTERNATE_BOM_ITEM	
TBOM_ALTERNATE_BOM_ITEM_A	
TBOM_CONSUMPTION_PLAN	
TBOM_CONSUMPTION_PLAN_A	
TBOM_PART_LIST	
TBOM_PART_LIST_A	
TBOM_PRODUCTION_PLAN	
TBOM_PRODUCTION_PLAN_A	
TBOM_REFERENCE_DESIGNATOR	
TEMPLATE_MFC	
TEMPLATE_MFC_A	
TEMPLATE_RECIPE	
TEMPLATE_RECIPE_A	
TEST_DEFINITION	

Back Next Finish

Select the tables you want to transfer in the **Tables** column and move them to the **Tables to be transferred** column by clicking [**>>**]. Select multiple tables by using the Shift or Ctrl keys.

IMPORTANT: If runtime tables are included in your table list (e.g., AT_<objectName>, DC_<objectName>, and UDA_<objectType>), the list you provide must also include the database table that provides the meta-data for the runtime table (e.g., APP_TABLE, DCS, and UDA).

3. Click [Next] to continue configuring the extract or [Finish] to accept the defaults and create the extract.
 - ▶ If you click [Next], go to [step 4](#).
 - ▶ If you click [Finish], go to [step 11](#).
4. If you click [Next], the following dialog displays.

Figure 10-3: Live Transfer Extract From Production (92Prod) to ODS (92ODS)

Live Transfer Extract From Production (92Prod) to ODS (92ODS)

Site Extract Manager Name:

Database Transaction Timeout:

Max Number of PIDs per Extract:

Message Delivery Timeout:

Resend Timeout:

Max Number of Rows per Message:

Maximum Extract Timeout:

5. Define the following properties:
 - ▶ **Site Extract Manager Name:** enter name of the SEM used by this Live Transfer configuration. The default is SEM.
 - ▶ **Database Transaction Timeout:** select the database transaction timeout used by the SEM to determine when there will be no further changes to the database records for a given PID value. Choose from 3 Minutes, 5 Minutes (default), and 10 Minutes.

NOTE: This value should be set to a value equal to the application server transaction timeout value or greater. For example, the default transaction timeout value for JBoss is 300 seconds (five minutes).

Therefore, set the Database Transaction Timeout value to 10 minutes. See your application server documentation for the default transaction timeout value for your application server.

IMPORTANT: Extract data will be dropped if the loader cannot finish loading data during the timeout period. If you are not sure what the timeout period should be, set the value to 0.

- **Max Number of PIDs per Extract:** select the maximum number of PIDs to include in a single extraction. Choose from System Defined (default), 1, 3, 5, and 10. The value for System Defined is set to $(\text{Database Transaction Timeout} / 30000 \text{ milliseconds}) + 2$.

This parameter should only be used when you have not run a transfer in a long time and need to get the data in the source and destination databases in line. This property limits the number of PIDs extracted in order to limit the transfer time.

For example, the default Database Transaction Timeout value is five minutes, which the system converts to 300,000 milliseconds for this calculation. Therefore, the default Max Number of PIDs per Extract size is 12 because $(300,000 / 30,000) + 2 = 12$.

- **Message Delivery Timeout:** select the amount of time a message sent to the loader from an extractor is allowed to sit in the loader's queue before being received by the loader. Choose from Unlimited and System Defined (default), which defaults to one minute.

If a message is not received in this amount of time, then ActiveMQ drops the message, and it will not be delivered. Setting this value to Unlimited means that a message sent to the loader will sit in the loader's queue indefinitely, which may cause issues with Live Transfer. For example, if the destination database is shut down unexpectedly, the extractor will be on hold indefinitely waiting for a response from the loader. Because of this, setting this value to Unlimited is not recommended.

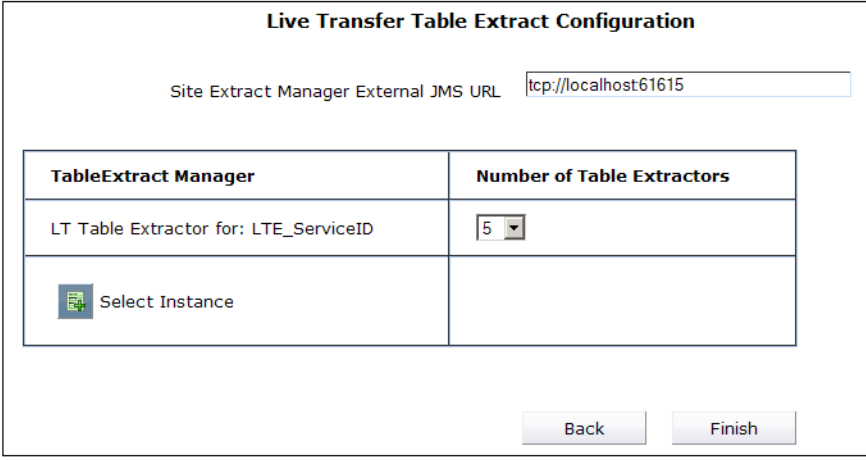
- **Resend Timeout:** select the amount of time that the extractor will wait for a message sent to the loader to be responded to before re-sending the message. Choose from 3 Minutes (default), 5 Minutes, and 10 Minutes.
- **Max Number of Rows per Message:** select the maximum number of rows allowed in a single message. Choose from 50, 250, 500 (default), and 1000.
- **Maximum Extract Timeout:** select the amount of time that must elapse when extracting a single PID before the SEM will stop and not issue any new extract requests. Choose from Unlimited (default), 30 minutes, and 60 minutes.

When the time elapses, a warning is issued informing the user that the extract for a specific PID has not been completed in the maximum amount


of time allotted. If this property is set to Unlimited (the default), then there is no maximum amount of time allotted for the extract, and the extract is not stopped regardless of how long it takes to extract a certain PID.

6. Click [Next] to continue configuring the extract or [Finish] to accept the defaults and create the extract.
 - ▶ If you click [Next], go to [step 7](#).
 - ▶ If you click [Finish], go to [step 11](#).
7. If you click [Next], the following dialog displays.

Figure 10-4: Live Transfer Table Extract Configuration






The dialog box is titled "Live Transfer Table Extract Configuration". It contains a text field for "Site Extract Manager External JMS URL" with the value "tcp://localhost61615". Below this is a table with two columns: "TableExtract Manager" and "Number of Table Extractors". The table has three rows: the first row is a header, the second row shows "LT Table Extractor for: LTE_ServiceID" with a dropdown menu set to "5", and the third row shows a "Select Instance" icon. At the bottom right are "Back" and "Finish" buttons.

TableExtract Manager	Number of Table Extractors
LT Table Extractor for: LTE_ServiceID	5
 Select Instance	

8. Define the **Site Extract Manager External JMS URL** property, which defines the connection information required to connect to the ActiveMQ broker embedded in the SEM. If the TEM is being run within the same JVM as the SEM, then set this property to `vm://localhost`. If the TEM is being run on a different JVM than the SEM, set this value to the same value specified for the **JMS Connection URL** property in the LTE's configuration ([step 2](#)).
9. Select the number of Table Extractor instances single TEM can host at one time. Choose from 1, 5 (default), 10, and 20. Note that the more Table Extractor instances you have, the more resources you will need.
10. Click the **Select Instance** icon to add TEMs. This list is populated by the TEMs' service ID names as defined in their *AdminedService.properties* file. See "Configuring Live Transfer Installations" on page 71 for details.
11. Click [Finish]. The configuration displays in the home page in the Live Transfer box.

Figure 10-5: Live Transfer Instance

Live Transfer		
Source Database	Destination Database	Last Processed Time
92Prod	 Configure 92ODS	
 New Extract	 New Destination	

Defining the Destination Database

To define the destination database:

1. In the Live Transfer box on the home page, do one of the following:
 - Click the **New Destination** icon if you are configuring the destination database first.
 - Click the **Configure <ODS_Name>** icon if you have already configured the source database.

The Live Transfer Load for ODS screen displays.

Figure 10-6: Live Transfer for Load ODS


Live Transfer Load For ODS (92ODS)

Destination Database

92ODS

Source Databases

☐ 92ODS
☒ 92Prod

Table Loader Manager	Number of Loaders
TLM_ServiceID	5
 Select Instance	

JMS Connection URL

tcp://localhost61616

Test

JMS User

JMS Password

PID Update Period

1 Minute

Back

Finish

2. Define the following properties:

- ▶ **Destination Database:** select the destination database from the list of registered ODS databases. The names listed are the names you defined when registering the database. They are not the actual database names.
- ▶ **Source Databases:** select the Production database(s) that will transfer data to the selected destination database. The names listed are the names you defined when registering the database. They are not the actual database names. If you defined the source database first, it will be selected here.
- ▶ **JMS Connection URL:** The connection information required to connect to the appropriate ActiveMQ broker in order to communicate with the extractors. The default ActiveMQ port is 61616. Click [Test] to test the JMS connection.
- ▶ **JMS User:** enter the user name for a user who has privileges on the ActiveMQ broker. If a user name is not specified, the system will use ActiveMQ's default user name, which is *system*. Depending on your configuration, this property may be optional.
- ▶ **JMS Password:** enter the password corresponding to the user name defined in the **JMS User** property. If a password is not specified, the system will use ActiveMQ's default password, which is *manager*. Depending on your configuration, this property may be optional.
- ▶ **PID Update Period:** select the minimum amount of time between increments to the ODS's current PID value. Choose from 1 Minute (default), 5 Minutes, 10 Minutes, 30 Minutes, and 60 Minutes.

The TLM will only increment the target ODS's PID value if there are incoming messages from the SEM. If the SEM is not sending data to the TLM, the TLM will not increment the ODS's current PID value.

3. Click the **Select Instance** icon to define a TLM. At least one TLM must be defined. This list is populated by the TLMs' service ID names as defined in their *AdminedService.properties* file. See “[Configuring Live Transfer Installations](#)” on page 71 for details.
The number of Table Loader instances a single TLM can host at one time. The default value is 5.
4. Click [Finish]. The configuration displays in the home page in the Live Transfer box.

Creating Instances from Existing Configurations

If you already have a complete Live Transfer configuration defined, you can configure additional source databases to transfer to the destination database defined for the existing configuration by performing the following steps.

1. From the FTPC Administrator home page, click the source database. The Live Transfer for Load ODS screen displays (see [Figure 10-6 on page 152](#)). Note the following:

- ▶ The ODS database configured for the existing Live Transfer configuration is selected.
- ▶ The Destination Database property is disabled.
- ▶ The source database from the existing Live Transfer configuration is selected by default in the Source Database property

See “[Defining the Destination Database](#)” on page 152 for details on the Live Transfer for Load ODS screen

2. In the Source Database property, select the additional Production databases that will transfer their data to the selected ODS.
3. Click [Finish]. The new configurations are displayed on the home page.

Editing Live Transfer Configurations

To edit an existing Live Transfer configuration, click the configuration row in the Live Transfer box on the FTPC Administrator home page and then select one of the following:

- ☐ To edit the extract configuration, click **Edit Extract Configuration** in the left panel. The Live Transfer Extract screen displays (see [Figure 10-1](#)). Follow the instructions in “[Defining the Source Database](#)” on page 146 to update your source database.
- ☐ To edit the load configuration, click **Edit Load Configuration** in the left panel. The Live Transfer for Load ODS screen displays (see [Figure 10-6](#)). Follow the instructions in “[Defining the Destination Database](#)” on page 152 to update your destination database.

NOTE: When editing a Live Transfer configuration, you cannot change the source and destination databases.

Sending Live Transfer Configurations

Configuration files for the configured Live Transfer components are written to `<FTPC_install>\FTPCAdminHome\adminData\<component>`. By default, this location is `C:\Rockwell\PO<version>.<build>\FTPCAdminHome\adminData\<component>`.

The source database configuration properties are stored in `...\<LTE_service_id>\current\SiteExtractManager.properties` file. Additional TEM configuration properties are stored in the `TableExtractManager.properties` file in the same location. The destination database configuration properties are stored in `...\<TLM_service_id>\current\TableLoaderManager.properties`.

Configuration files for new or edited components are stored in the **next** subdirectory, which is located at the same level as the **current** subdirectory and has the same subdirectory structure. The **current** subdirectory always reflects the Live Transfer components' current configurations on their respective systems.

A new Live Transfer service will only have property files in the **next** subdirectory. Once the configuration files are successfully sent to the target components, the **next** configurations are moved to the corresponding subdirectory in the **current** subdirectory and removed from the **next** subdirectory.

When you modify an existing Live Transfer configuration, its property files are contained in both its **current** and **next** subdirectories. Once the configuration files are successfully sent to the target components, the **next** configurations are moved to the corresponding **current** subdirectory and removed from the **next** subdirectory.

After you have created new configurations or updated existing ones, send the new and updated information to the Live Transfer service components (SEM, TEM, TLM).

1. Click the Live Transfer configuration in the Live Transfer box on the home page.
2. Select **Send Configuration**. The following screen displays.

Figure 10-7: Live Transfer Send Configuration From Production to ODS

Live Transfer Send Configuration From Production (pqj_cpjprod) to ODS (pqj_cpjprod1)

Send Configuration To All Services

Service	Configuration Status	Service Status	
SEM ID: LTE_ServiceID	Changes Pending	Available	Send
TLM ID: TLM_ServiceID	Changes Pending	Available	Send

Back

3. You have the option of sending all the updates or updates for specific services:
 - ▶ To send all the updates, click [Send Configuration to All Services]. This will move the configurations currently saved in the **next** subdirectory of each service to the corresponding **current** subdirectory, and the configurations in the **next** subdirectory are removed.
 - ▶ To send updates to specific services, click [Send] in the row of the service you want to update. This will move the configurations currently saved in the **next** subdirectory of the specific service to the corresponding **current** subdirectory, and the configuration in the **next** subdirectory is removed.

When the configuration has been successfully sent, the Send button will be disabled.

A dialog displays notifying you that the Live Transfer service will be unavailable while the configuration is being sent.

Using the Comparison Tools

Installed with Live Transfer are two comparison tools that allow you to compare records between the source database and the target database. These tools can be used to verify that Live Transfer is functioning properly.

The comparison tools are described below:

- ☐ **LTPIDCompare:** compares selected PIDs. You can specify the tables and PIDs to be compared.
- ☐ **LTSampleCompare:** compares a random sample of rows. You can specify the tables and percentage of PIDs to be compared.

IMPORTANT: Before running any comparison tool, ensure that FTPC and Live Transfer are stopped.

Using LTPIDCompare

To run the LTPIDCompare tool, perform the following steps:

1. Stop the FTPC and Live Transfer services if they are running.

TIP: This tool can be used while Live Transfer and FTPC are running, but make sure that the system will not be changing rows within the selected PIDs and that Live Transfer has completed processing of the specified PIDs.

2. Open a command prompt. Change the directory to *<LT_install>*, where *LT_install* is the Live Transfer installation directory. The default installation directory on Windows is C:\Rockwell\FTPC_LiveTransfer.
3. Enter the following on a single line:

```
<win or linux>\java\jre\bin\java -cp lib\*
com.rockwell.datatransfer.tools.LTPIDCompare <extractHome>
<loaderHome> <silentOperation> <PIDList> <specificTables>
```

The parameters are described below:

- ❖ **extractHome:** enter the parent directory of the conf folder containing the SiteExtractManager.properties file. By default, this directory is *<win or linux>\SiteExtractManager*.

- ❖ **loaderHome**: enter the parent directory of the conf folder containing the TableLoaderManager.properties file. By default, this directory is *<win or linux>\TableLoaderManager*.
 - ❖ **silentOperation**: enter either **true** or **false**. If set to true, messages are only printed if there are problems. If set to false, all messages are printed.
 - ❖ **PIDList**: enter a comma delimited list of PIDs to compare.
 - ❖ **specificTables**: enter a comma delimited list of tables, or enter **NONE**. If tables are specified, only those tables will be tested. Else, all tables will be tested.
4. Run the command to begin the comparison.

Using LTSampleCompare

To run the LTSampleCompare tool, perform the following steps:

1. Stop the FTPC and Live Transfer services if they are running.
2. Open a command prompt. Change the directory to *<LT_install>*, where *LT_install* is the Live Transfer installation directory. The default installation directory on Windows is C:\Rockwell\FTPC_LiveTransfer.
3. Enter the following on a single line:

```
<win or linux>\java\jre\bin\java -cp lib\*
com.rockwell.datatransfer.tools.LTSampleCompare <extractHome>
<loaderHome> <silentOperation> <samplePercent> <specificTables>
```

The parameters are described below:

- ❖ **extractHome**: enter the parent directory of the conf folder containing the SiteExtractManager.properties file. By default, this directory is *<win or linux>\SiteExtractManager*.
- ❖ **loaderHome**: enter the parent directory of the conf folder containing the TableLoaderManager.properties file. By default, this directory is *<win or linux>\TableLoaderManager*.
- ❖ **silentOperation**: enter either **true** or **false**. If set to true, messages are only printed if there are problems. If set to false, all messages are printed.
- ❖ **samplePercent**: enter an integer between 1 and 100. This sets the percentage of PIDs to test, per table.
- ❖ **specificTables**: enter a comma delimited list of tables, or enter **NONE**. If tables are specified, only those tables will be tested. Else, all tables will be tested.

Run the command to begin the comparison.

Starting and Stopping Live Transfer

To start Live Transfer:

- 1. Click the Live Transfer configuration in the Live Transfer box on the home page. The Live Transfer From Production to ODS screen displays. Depending on your extract configuration, the contents of the screen will vary:
 - If the configuration is extracting all tables, a screen similar to Figure 10-8 displays.
 - If the configuration is extracting a specific list of tables, a screen similar to Figure 10-9 displays.

Figure 10-8: Live Transfer From Production to ODS Screen (Extract All)

Live Transfer From Production (92Prod) to ODS (92ODS)

Status

Refresh

Process		Interval		Status	Message
StartTime	EndTime	StartTime	EndTime		
None	None	None	None	None	None

Services

Refresh

Process	Status		
SEM:LTE_ServiceID	Available	Start	Stop
TLM:TLM_ServiceID	Available	Start	Stop

Figure 10-9: Live Transfer From Production to ODS Screen (Extract List)

Live Transfer From Production (92Prod) to ODS (92ODS)

Status

Refresh

Table Name	Last Interval Loaded
BATCH	None
BATCH_A	None
SUBLOT	None
SUBLOT_A	None
SUBLOT_RELATION	None
SUBLOT_RELATION_A	None
SUBLOT_STORAGE_HISTORY	None

Service

Refresh

Process	Status		
SEM:LTE_ServiceID	Available	Start	Stop
TLM:TLM_ServiceID	Available	Start	Stop

2. Click [Start] for each service to start Live Transfer.
3. Click [Stop] for each service to stop Live Transfer.

Configuring Live Transfer Logging

To configure Live Transfer logging:

1. Open the *logging.properties* file located at `<LT_Install>\<either_win_or_linux>\LiveTransferExtractor\conf` where `<LT_Install>` is where you installed Live Transfer.

2. Locate and uncomment the following lines.

```
com.rockwell.datatransfer.extract.manager.SiteExtractManager.  
level = FINEST  
  
com.rockwell.datatransfer.tableextract.manager.TableExtractor  
Manager.level = FINEST  
  
com.rockwell.datatransfer.tableloader.manager.TableLoaderMana  
ger.level = FINEST
```

3. Enter the level of logging you want for each Live Transfer component. The supported levels in descending order are as follows:
 - SEVERE
 - WARNING
 - INFO
 - CONFIG
 - FINE
 - FINER
 - FINEST
4. Save and close the file.

Configuring ActiveMQ

By default, no additional configuration is required for ActiveMQ. However, depending on your environment, you may want to perform the following configurations.

Go to <http://activemq.apache.org/> for more information on ActiveMQ.

Configuring a Separate ActiveMQ Cluster

ActiveMQ is clustered by default. If you want to configure a separate ActiveMQ broker or network of brokers, follow the instructions at <http://activemq.apache.org/shared-file-system-master-slave.html> to set up your shared file system master slave environment. Please use the configuration where the persistence adapter is defined by the KahaDB directory.

Configuring ActiveMQ for a Firewall Environment

To set up ActiveMQ for a firewall environment where Live Transfer is in a demilitarized zone (DMZ) and ActiveMQ is outside the DMZ, use an HTTP transport connector. Perform the following steps:

1. Open the *activemq.xml* file in a text editor. This file is located in `<LT_install>\<either_win_or_linux>\JMS-Broker\apache-activemq-<version>\conf`.

2. Find the transport connectors section and add an HTTP transport connector. For example:

```
<transportConnectors>
<transportConnector name="ssl" uri="ssl://localhost:61617"/>
<transportConnector name="stomp" uri="stomp://localhost:61613"/>
<transportConnector name="xmpp" uri="xmpp://localhost:61222"/>
<transportConnector name="http" uri="http://<hostname>:<port>"/>
</transportConnectors>
```

3. Save and close the *activemq.xml* file.
4. From the machine that will be running the SEM, go to *<LT_install>\<either_win_or_linux>\LiveTransferExtractor\conf* and open the *SiteExtractManager.properties* file in a text editor.
5. Set the **loaderBrokerURL** property to the URL you set in Step 2 to connect to ActiveMQ using the HTTP transport connector.
6. Save and close the *SiteExtractManager.properties* file.
7. From the machine that will be running the TLM, go to *<LT_install>\<either_win_or_linux>\TableLoaderManager\conf* and open the *TableLoaderManager.properties* file in a text editor.
8. Set the **loaderBrokerURL** property to the URL you set in Step 2 to connect to ActiveMQ using the HTTP transport connector.
9. Save and close the *TableLoaderManager.properties* file.

Configuring ActiveMQ Monitoring

Assuming that you have the Jetty configured in your *activemq.xml* file located in *<LT_install>\<either_win_or_linux>\JMS-Broker\apache-activemq-<version>\conf*, you can use a web browser to monitor the queues in an ActiveMQ broker instance.

IMPORTANT: The embedded broker housed in the SEM instance does not run the Jetty instance and, therefore, is not exposed for monitoring. To monitor the traffic between the SEM and TEM, contact Technical Support for additional configuration.

The following sample configuration in the *activemq.xml* file defines the URL *http://localhost:8161/admin* to be used to monitor the ActiveMQ queues.

```
<!-- An embedded servlet engine for serving up the Admin console
-->

<bean id="Connector"
class="org.eclipse.jetty.server.ServerConnector">
```

```

    <constructor-arg ref="Server" />
    <!-- see the jettyPort bean -->
    <property name="host" value
      ="#{systemProperties['jetty.host']}" />
  </bean>
  <property name="handlers">
    <list>
      <ref bean="rewriteHandler"/>
      <bean class="org.eclipse.jetty.webapp.webAppContext">
        <property name="contextPath" value="/admin"/>
        <property name="resourceBase"
          value="${activemq.home}/webapps/admin"/>
        <property name="logUrlonStart" value="false"/>
      </bean>
      ...
      ...
      <bean id="defaultHandler"
        class="org.eclipse.jetty.server.handler.DefaultHandler">
        <property name="serveIcon" value="false"/>
      </bean>
    </list>
  </property>

```

To monitor the queues, browse to the defined URL and click the *Queues* link. This will show the queues in the broker with the number of messages pending, sent, and delivered as well as the present number of consumers.

Figure 10-10: ActiveMQ Queues

Name	Number Of Pending Messages	Number Of Consumers	Messages Enqueued	Messages Dequeued	Views	Operations
FTPCAdminLog_192.168.79.129	0	0	0	0	Browse Active Consumers Active Producers	Send To Purge Delete
FTPCAdminLog_192.168.79.155	0	0	0	0	Browse Active Consumers Active Producers	Send To Purge Delete
FTPCAdminLog_192.168.79.174	0	0	0	0	Browse Active Consumers Active Producers	Send To Purge Delete
FTPCAdminLog_192.168.79.206	0	1	153	153	Browse Active Consumers Active Producers	Send To Purge Delete
LoaderBlobDataQueue_99f816270d474f1397668c8ce89...	0	5	3	3	Browse Active Consumers Active Producers	Send To Purge Delete
LoaderDefinitionDataQueue_99f816270d474f1397668...	0	5	110	110	Browse Active Consumers Active Producers	Send To Purge Delete
LoaderNormalDataQueue_99f816270d474f1397668c8ce...	0	5	300	300	Browse Active Consumers Active Producers	Send To Purge Delete
QM5vcEvaluation	0	0	0	0	Browse Active Consumers Active Producers	Send To Purge Delete
RecelptQueue_SEH_9066EBE5BFD409391302912351F5A32	0	1	414	414	Browse Active Consumers Active Producers	Send To Purge Delete

Configuring Live Transfer with Linux

If you will be running Live Transfer on Linux, additional configuration in the Java Service Wrapper (*wrapper.conf*) is required.

1. Open the *wrapper.conf* file for the extractor located in the following directory by default:

```
<LT_install>\linux\LiveTransferExtractor\conf
```

2. Locate the following line:

```
wrapper.java.additional.1=
-Djava.ext.dirs=../../../../lib;../../../../java/jre/lib/ext;../../../../jav
a/lib/ext;
```

3. Edit it so that it looks like the following:

```
wrapper.java.additional.1=
-Djava.ext.dirs=../../../../lib:%JAVA_HOME%/jre/lib/ext:%JAVA_HO
ME%/lib/ext;
```

4. Save and close the file.
5. Open the *wrapper.conf* file for the loader located in the following directory by default:

<LT_install>\linux\TableLoaderManager\conf

6. Repeat step 2 through step 4.

Setting the Maximum Memory

Set the maximum memory value in the Linux Java Service Wrappers. This value is the maximum amount of memory that Live Transfer (both the extractors and loaders) can use. If this amount of memory is not available when Live Transfer is run, an out-of-memory exception is thrown.

The maximum memory setting does not need to be the same for all the wrapper files. For example, if you are increasing the number of loaders only, then update the maximum memory setting for the TLM's wrapper file only as appropriate.

The location of the files to be edited are as follows:

- ❑ If the SEM and TEM are running in the same JVM, edit the *wrapper.conf* file located at <LT_install>/linux/LiveTransferExtractor/conf/.
- ❑ If the SEM and TEM are running in different JVMs, edit the *wrapper.conf* files located at <LT_install>/linux/SiteExtractManager/conf/ for the SEM and <LT_install>/linux/TableExtractManager/conf/ for the TEM.
- ❑ Edit the *wrapper.conf* file located at <LT_install>/linux/TableLoaderManager/conf/ for the TLM.

In each *wrapper.conf* file, uncomment the following line and set the value to at least 512, which is the recommended value to support five extractors or loaders.

```
#wrapper.java.maxmemory=64
```

Change this value accordingly as Live Transfer's memory consumption increases. For example, if you have 10 table loaders configured, change the maximum memory value in the TLM's *wrapper.conf* file to 1024 (512 * 2) to allocate more memory to the table loaders in order to avoid an out-of-memory exception.

Configuring the Java Service Wrapper (Optional)

NOTE: This section only applies if you are running Live Transfer in a Windows environment.

You have the option to configure the following properties in the Java Service Wrapper file:

1. Open the *wrapper.conf* file located in the following directory by default:
 <LT_install>\<either_win_or_linux>\LiveTransferExtractor\conf
2. Check the following properties to make sure they are appropriate for your environment.

- The Java command:

```
wrapper.java.command=%JAVA_HOME%/bin/java
```

- Name of the service:

```
wrapper.ntservice.name=Rockwell FTPC LiveTransferExtractor  
10.0.120999
```

- Name of the service that displays in Control Panel's Services interface:

```
wrapper.ntservice.displayname=Rockwell FTPC  
LiveTransferExtractor 10.0.120999
```

- The JVM exit timeout property. This property defines the number of seconds allowed between the time that the JVM reports that it is stopped and the time that the JVM process actually terminates. This will take effect in the beforeStop and afterStop events:

```
wrapper.jvm_exit.timeout=15
```

3. If necessary, make any changes to the properties to fit your environment.
4. Save and close the file.

Deployment Options and Recommendations

The Live Transfer environment can be set up in multiple ways, but please note the following recommendations.

Extraction Recommendations

Live Transfer Extractor (LTE) instances should be used in a 1-to-1 relationship with source databases. The LTE service houses an instance of the SEM and an instance of the TEM in a single JVM to maximize the efficiency of the ActiveMQ JMS communication between the TEM and SEM.

If Live Transfer is not extracting data quickly enough from the source database and increasing the number of table extractors in the TEM instance does not remedy the problem, then the system provides for a scaling of extract capacity by adding additional TEM instances running on additional machines. Because increasing the number of table extractors creates additional threads, this may not solve the issue of slow extractions because the additional task-switching involved may increase the amount of required resources on the system. In this case, another TEM instance may be required. These additional TEM instances will process the extract messages generated by the SEM instance housed in the LTE service and process the load receipts returned by the table loaders housed in the TLM instance(s).

The default number of five table extractors per TEM is based on the following estimations with Live Transfer running in a steady state:

- ❑ 50 tables are regularly updated by a running application, and each table extract should take about three seconds. Therefore, extracting information from 50 tables should take about 150 “working seconds.”
- ❑ The extract frequency is 30 seconds. 150 “working seconds” for the extractions at a 30 “real seconds” frequency would require five table extractors to perform the work.

If the number of tables extracted, extract time, and frequency for your environment differs from these estimations, then configure your number of required table extractors accordingly.

Loader Recommendations

TLM instances are used in an N-to-1 relationship with target databases because the system is more likely to have a bottleneck during the load process because multiple source databases can be loaded into a single target database. Because of this, the system has been designed for loader scalability in the following ways.

- ❑ Adjust the number of table loaders housed in a TLM instance. The number of table loader instances is defined by the **Number of Loaders** property in the Live Transfer for Load ODS screen (Figure 10-6 on page 152). The default number of loaders is five, which was determined by the same estimates used for the default number of extractors defined in the previous section.

In general, the number of table loaders should be the same as the total number of table extractors sending data to the target database. Therefore, if there are three sources databases with five table extractors each, the expectation is that 15 table loaders will be required.

- ❑ Add additional TLM instances running on additional machines if a single instance of the TLM is unable to support the required number of table loaders. You will probably need additional TLM instances if you have observed that your source database server, extractors, and target database server are running at a steady rate, but the data to be loaded is piling up in the loader queue and not being loaded in a timely manner. If multiple instances of the TLM are required, then FTPC Admin will ensure that the multiple TLM instances will have the same configurations with the possible exception of the **Number of Loaders** property.

ActiveMQ Recommendations

The Live Transfer installer provides a service for running ActiveMQ. If you already have ActiveMQ installed, please use the Rockwell-provided ActiveMQ package for FTPC purposes.

We recommend that you configure a single ActiveMQ instance that will be used by all of the FTPC components such as Live Transfer, Purge, and the customer application (through the use of Shop Operations Server and Shop Operations

Client components). In order to avoid this single instance becoming a single point of failure, the instance should be a failover load-balanced ActiveMQ cluster. Where the ActiveMQ service instances should be located depends on the expected load:

- ❑ In light load installations, the ActiveMQ service instances may be run on the same machines where the LTE and TLM instances are running.
- ❑ In heavy load installations, the ActiveMQ service instances should be run on machines that are dedicated to providing messaging service.

Multiple “Network Remote” Locations

If Live Transfer is communicating between databases located in different geographic locations that have firewalls preventing native ActiveMQ connections between the sites, we recommend that you configure HTTP transports for your ActiveMQ broker instance to allow remote connections.

You can also build a network of brokers with HTTP “network connectors” tying the brokers separated by firewalls apart. However, the installation will have to include the list of queues to be forwarded, which creates a higher administration burden.

Deployment Scenarios

The scenarios in this section assume the following:

- ❑ **ActiveMQ Presence:** For each of the scenarios, it is assumed that an ActiveMQ instance is available to the LTE and TLM instances required for Live Transfer. The ActiveMQ instance will be hosted on either the same machine(s) being used for Live Transfer or on dedicated machines depending on the scenario.
- ❑ **Light Load vs. Heavy Load:** For the purposes of these scenarios, Production database growth per day of 1 GB or greater is defined as a heavy load. Production database growth per day of 500 MB or less is defined as a light load. A load between 500 MB and 1 GB can be either a light or heavy load.
- ❑ **Network Local vs. Network Remote:** For the purposes of these scenarios, a network local is defined as a network configuration in which a single server can make JDBC connections to both the source and target databases. A network remote connection is defined as a network configuration in which no single server can make JDBC connections to both the source and target databases.

This section covers the following Production database-to-Base ODS scenarios:

- ❑ “Light Load/Network Local” on page 168
- ❑ “Heavy Load/Network Local” on page 168
- ❑ “Load Independent/Network Remote” on page 169

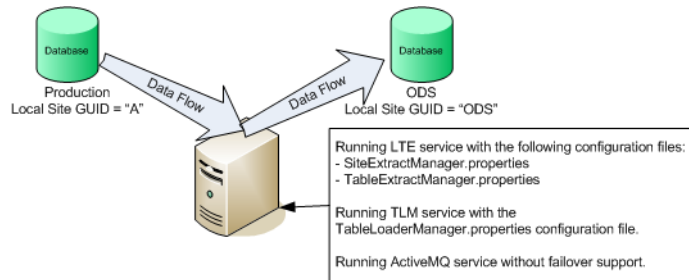
- ❑ “Heavy Load/Network Local” on page 170

Light Load/Network Local

This scenario assumes the following:

- ❑ A single machine is running both the LTE and TLM services.
- ❑ A single instance of the ActiveMQ service is running on the same machine.
- ❑ The ODS is being loaded with data from a single Production database with a relatively low amount of transactions.

Figure 10-11: Light Load/Local Network Scenario



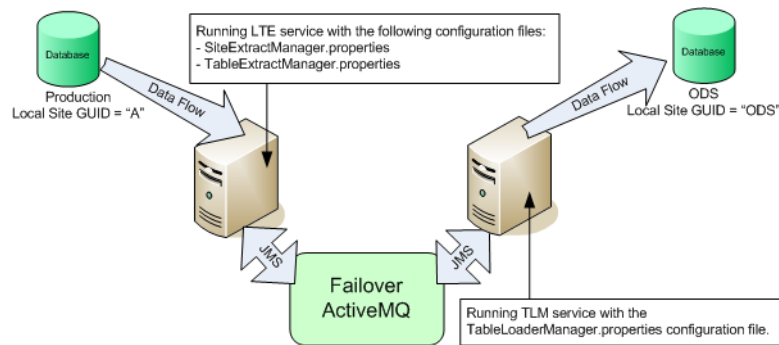
In this configuration, the following properties should all have the same value [for example: failover:(tcp://localhost:61616)]:

- ❑ LTE’s JMS Connection URL
- ❑ TEM’s Site Extract Manager External JMS URL
- ❑ TLM’s JMS Connection URL

Heavy Load/Network Local

This scenario assumes the following:

- ❑ One machine is running the LTE service and another machine is running the TLM service.
- ❑ The ODS is being loaded with data from a single database, but the amount of data per day is greater than 1 GB, and there is a heavy reporting requirement being satisfied by the ODS.

Figure 10-12: Heavy Load/Local Network Scenario

In this configuration, the following properties should all have the same value (for example: failover:||(tcp://amqprimary:61616),tcp://amqsecondary:61616)?initialReconnectDelay=100):

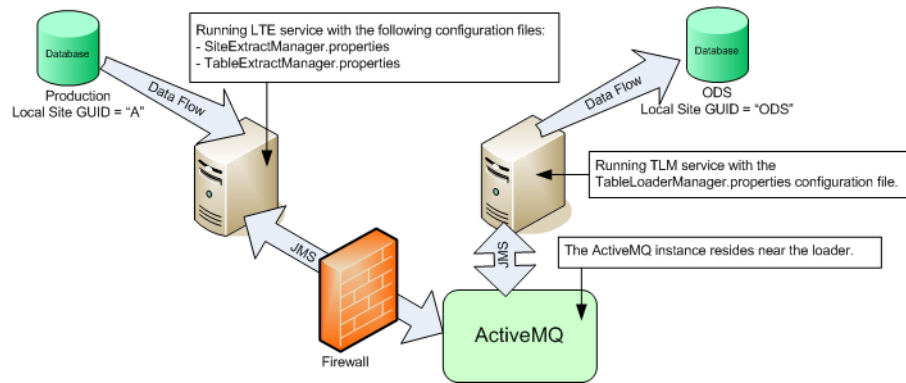
- ☐ LTE's JMS Connection URL
- ☐ TEM's Site Extract Manager External JMS URL
- ☐ TLM's JMS Connection URL

In addition, the number of extractors and loaders can be increased to accommodate the heavy load.

Load Independent/Network Remote

This scenario assumes the following:

- ☐ No one server can make JDBC connections to both databases, so two machines are required.
- ☐ One machine that is in close proximity to the source database server so that it can make a JDBC connection to the source database is running the LTE service
- ☐ Another machine that is in close proximity to the target database server so that it can make a JDBC connection to the target database is running the TLM service.
- ☐ A firewall located such that an HTTP transport needs to be used by the LTE instance. In this arrangement, the ActiveMQ instance is running in close proximity to the TLM instance and the LTE components are communicating by HTTP transport. This is not a “network of brokers” configuration.

Figure 10-13: Load Independent/Remote Network Scenario

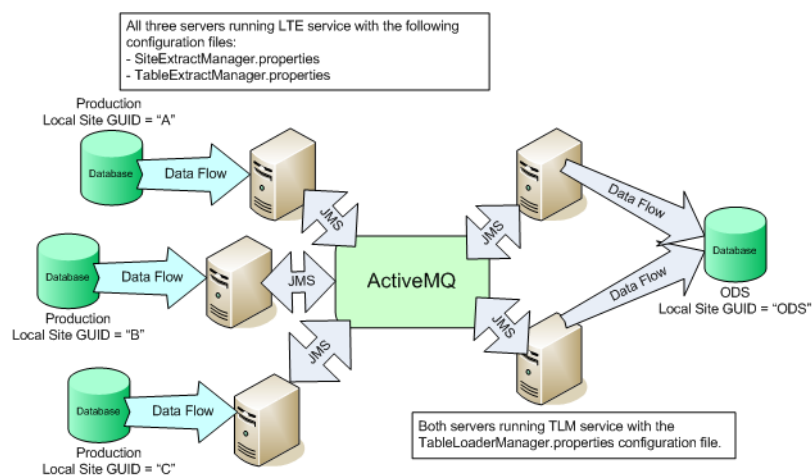
In this configuration, the following properties are set as follows:

- ❑ The LTE's JMS Connection URL and TEM's Site Extract Manager External JMS URL are configured as an HTTP transport to the ActiveMQ instance (for example: `http://activemq:80`)
- ❑ The TLM's JMS Connection URL is defined as a normal TCP URL because the ActiveMQ instance is local to the TLM instance [for example: `failover:(tcp://activemq:61616)`].

Heavy Load/Network Local

This scenario assumes that there are five machines:

- ❑ An LTE service is running on three machines extracting from three Production databases.
- ❑ Each LTE instance requires 10 table extractors.
- ❑ A TLM service is running on two machines with each machine running 15 table loaders because one machine cannot support all 30 table loaders.

Figure 10-14: Heavy Load/Local Network Scenario

In this configuration, the following properties are defined as follows:

- ❑ The JMS URL for A, B, and C are the same [for example: failover:(tcp://activemq:61616)].
- ❑ The number of extractors is increased from five to ten in the extract configuration for Sites A, B, and C. Therefore, there are 30 extractors total.
- ❑ The TLM's JMS Connection URL for both loaders are the same [for example: failover:(tcp://activemq:61616)].
- ❑ The number of loaders is increased from five to 15 in the loader configuration for both TLMs. Therefore, there are 30 extractors total, which allows them to handle the extractions from three source databases.

Configuring Purge

In this chapter

- ❑ **Creating a Purge Job** 174
- ❑ **Configuring the Purge Service** 180
- ❑ **Sending Purge Configurations** 183
- ❑ **Starting and Stopping Purge Jobs** 184
 - Running A Purge Job Manually 185
- ❑ **Viewing Purge Statuses** 185
- ❑ **Updating Data Collection Maps (Status-Based Only)** 186

All databases that will be purged must be registered in FTPC Administrator before they can be selected in a Purge configuration. See “[Registering a Database](#)” on page 101 for details.

IMPORTANT: Once Purge is started, changes the configuration will be ignored. To change any of the configurations, you must first stop Purge, make the changes, and then restart Purge. When Purge is restarted, it will pick up where it left off before it was stopped as if there had been a power failure. See “[Handling a Communication Loss](#)” on page 197 for details.


When you access the FTPC Administrator home page, click the **Search Purge Services** link to find any existing Purge services. If a service is found, it will have one of the following statuses:

- ☐ **Running:** both the wrapper and the service are running.
- ☐ **Available:** the wrapper is running, but the service itself is not.
- ☐ **Unavailable:** FTPC Administrator has found a service that had been running before, but neither the wrapper nor the service is currently running.

The service ID displayed is defined in Purge's AdminedService.properties file. See “[Configuring Purge Installations](#)” on page 84 for details.

Figure 11-1: Search Purge Services

Purge

 Search Purge Services

Service ID	Status
Purge_serviceID	Running
Purge_ServiceID	Unavailable

See “[Registering and Starting the Purge Service](#)” on page 85 for information on starting the Purge service.

Creating a Purge Job

To create a Purge job:

- On the FTPC home page, click the **Search Purge Services** link. The system will display any Purge services found. The ID displayed is defined in Purge’s *jms-local.config* file. See “[Configuring Data Management Installations](#)” on page 94 for details.
- Click on the Purge service ID. The Purge Service Summary screen displays.

Figure 11-2: Purge Service Summary Screen

Service Summary

Service Management

Start Stop Send Configuration

Service Job Management

Service Jobs

Purging Strategy	Database Purged	Database Analyzed

Edit Job Delete Job Create Job

Job Status For Selected Job

Data Category	Start PID	End PID	Job Status

Refresh Status Run Job Now Update Data Collection Map

3. In the Service Job Management box, click [Create Job]. The Database To Be Purged Screen displays.

Figure 11-3: Database To Be Purged Screen

Database To Be Purged

92_Prod

Back Next

4. Select the database to be purged from the drop-down menu and click [Next]. The Purge Strategy Screen displays.

Figure 11-4: Purge Strategy Screen

Purging Strategy

Purging Strategy Status Based

Database to be Analyzed 92_Prod

Back Next

5. Define the following properties and click [Next].

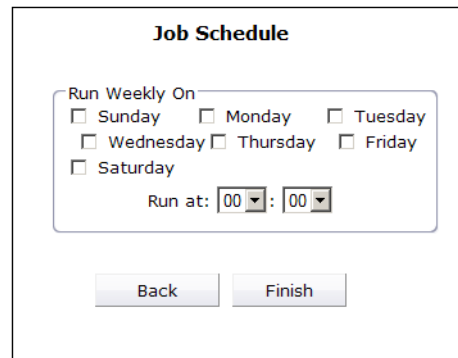
- ▶ If the database to be purged is a Production database, define the following properties:
 - ❖ **Purging Strategy:** select from either Status Based (default) or Time Based.
 - ❖ **Database to be Analyzed:** select that database where the marking will take place. This property is only visible when the purging strategy is Status Based.
 - Select the Production database itself if you want the marker to run on the Production database.
 - Select an ODS if you want the marker to run on the ODS.
 - ▶ If the database to be purged is an ODS, define the following properties:
 - ❖ **Purging Strategy:** Status Based is selected by default, and the property is disabled. Time Based is not an option.
 - ❖ **Production Database To Be Deleted:** select the Production database whose data will be deleted from the ODS for this Purge job.
6. In the Data Retention Screen, define the following properties:
- ▶ For Time-Based Purge jobs:
 - ❖ **TRX Record Deletion Range Size (Record):** enter the maximum number of tracked objects that can be loaded into memory. The default is 1000.
 - ❖ **Retention Period for Runtime Objects (days):** enter when to delete runtime objects. The default is 200 days.
 - ❖ **Retention Period for Test/DC Objects (days):** enter when to delete test and data collection objects. The default is 200 days.
 - ❖ **Retention Period for Transactions (days):** enter when to delete transactions. The default is 200 days.
 - ❖ **Purge Interval (PID):** enter the maximum number of PID intervals that will be examined during a marking session. The default is 14.
 - ▶ For Status-Based Purge jobs:
 - ❖ **TRX Record Deletion Range Size (Record):** enter how many rows per deletion statement are tagged during the marker process. The default is 1000.

For example, you want to delete all rows that have an *xfr_update_pid* value of 100. There are 10,000 entries with this *xfr_update_pid* value, and this property is set to 1000. When the marker starts tagging the rows with the PID value of 100, it will mark 1000 rows at a time. Therefore, 10 deletion statements will be issued instead of 10,000.

IMPORTANT: If this property is set too high, the TRX tables may become locked. Make sure you have enough resources to accommodate your configuration.

- ❖ **Job Duration Limit (mins):** enter how long (in minutes) each job will last. Setting this limit will allow an administrator to schedule jobs to start during a maintenance window and know that the job will stop running before the end of the window. The default, 0, means there is no limit. This property only displays if you are configuring a status-based Purge job that uses the ODS marker.
- ❖ **Retention Period for Batches (days):** enter when to delete batches. The default is 200 days. For more information on this retention period, see “[Batches](#)” on page 45. This property only displays if you are configuring a status-based Purge job that uses the ODS marker.
- ❖ **Retention Period for ObjectStateHistories (days):** enter when to delete non-current ObjectStateHistories for objects that are not subject to being purged. The default is 200 days. For more information on this retention period, see “[ObjectStateHistory](#)” on page 47. This property only applies if you are configuring a status-based Purge job that uses the ODS marker.
- ❖ **Retention Period for ProcessOrders (days):** enter when to delete process orders. The default is 200 days. For more information on this retention period, see “[Process Orders](#)” on page 47. This property only displays if you are configuring a status-based Purge job that uses the ODS marker.
- ❖ **Retention Period for Reports and Labels (days):** enter when to delete report and label objects. The default is 200 days. For more information on this retention period, see “[Reports and Labels](#)” on page 48. This property only displays if you are configuring a status-based Purge job that uses the ODS marker.
- ❖ **Retention Period for Stateless Runtime Objects (days):** enter when to delete stateless runtime objects. The default is 200 days. For more information on this retention period, see “[Stateless Runtime Objects](#)” on page 49.
- ❖ **Retention Period for Inventory Container History (days):** enter when to delete inventory container history objects. The default is 200 days. For more information on this retention period, see “[Inventory Container History](#)” on page 50.
- ❖ **Retention Period for Tracked Runtime Objects (days):** enter when to delete tracked runtime objects. The default is 200 days. For more information on this retention period, see “[Tracked Runtime Objects](#)” on page 51.

- ❖ **Retention Period for WorkFlow Objects (days):** enter when to delete workflow objects. The default is 200 days. For more information on this retention period, see “Workflows” on page 55.
 - ❖ **Retention Period for Consolidated Log (days):** enter when to delete consolidated log records. The default is 50 days. For more information on this retention period, see “The retention period for workflows can affect how tracked runtime objects are purged. The retention period for tracked runtime objects must be equal to or longer than the retention periods of stateless runtime objects and workflows. If this requirement is not met, the following error will result when the system attempts to purge tracked runtime objects: "Tracked object marker cannot run ahead of the stateless and workflow markers."” on page 55.
 - ❖ **Retention Period for Transactions (days):** enter when to delete transactions. The default is 200 days. For more information on this retention period, see “Transactions” on page 55.
 - ❖ **Retention Period for Pallet (days):** enter when to delete pallets. The default is 200 days. For more information on this retention period, see “Pallets” on page 56.
 - ❖ **Status for Purge:** Enter the status of the objects to be purged. Select one of the following statuses:
 - *Closed*: all tracked objects with the status Closed will be purged. This is the default.
 - *Shipped*: all tracked objects with the status Shipped or Closed will be purged.
 - *Finished*: all tracked objects with the status Finished, Shipped, or Closed will be purged.
 - ❖ **Purge Interval (PID):** enter the maximum number of PID intervals that will be examined during a marking session. The default is 14.
7. Click [Next] to schedule the Purge. The Job Schedule screen displays.
- If you do not want to define a schedule (for example, you will be starting the Purge job manually as needed), click [Finish].

Figure 11-5: Job Schedule Screen


Job Schedule

Run Weekly On

☐ Sunday ☐ Monday ☐ Tuesday

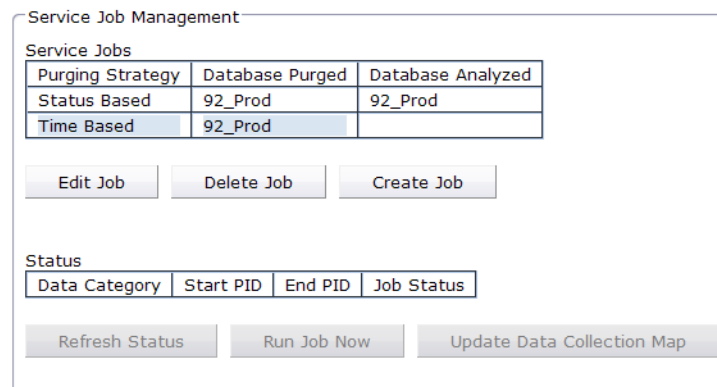
☐ Wednesday ☐ Thursday ☐ Friday

☐ Saturday

Run at: :

8. Enter the time when you want this Purge job to run and click [Finish]. Your configured Purge jobs are added to the Purge Service Summary page.

NOTE: A Purge job may be launched earlier than its scheduled time by up to the value of the wake-up-period, which is defined, in minutes, in the DSServerSchedule.xml file. (See “[Sending Purge Configurations](#)” on [page 183](#) for the file location.) For example, if the job is scheduled for 5:00 PM and the wake-up period is 15 minutes, the job may be started as early as 4:45.

Figure 11-6: Configured Purge Jobs


Service Job Management

Service Jobs

Purging Strategy	Database Purged	Database Analyzed
Status Based	92_Prod	92_Prod
Time Based	92_Prod	

Status

Data Category	Start PID	End PID	Job Status

What is displayed in the Database Analyzed column depends on the Purge job configuration:

- ▶ If the Purge job is a time-based job, no database is listed.
- ▶ If the Purge job is a status-based job that purges a Production database, the database where the marker is run is listed.
- ▶ If the Purge job is a status-based job that purges an ODS, the Production database whose data will be deleted from the ODS during this job is listed.

9. To edit a Purge job, select it and click [Edit Job].

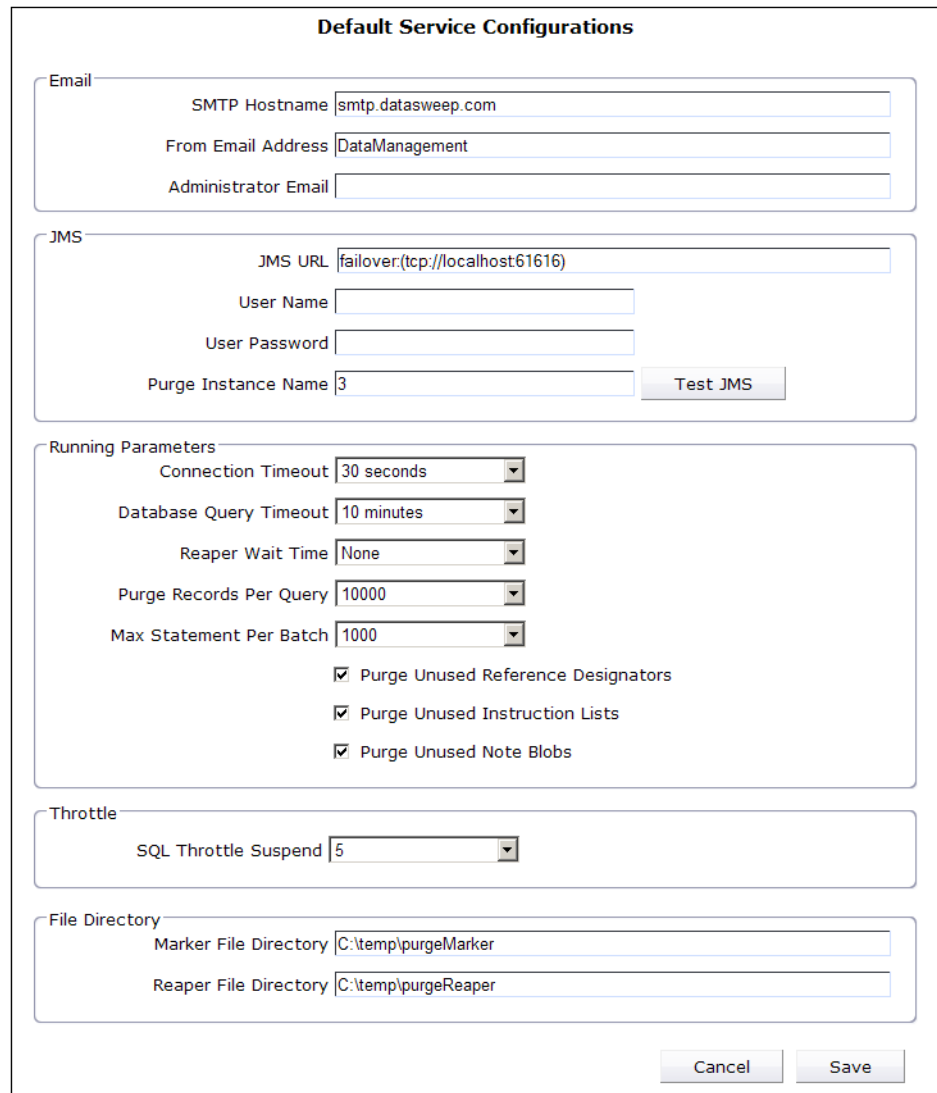
10. To delete a Purge Job, select it and click [Delete Job].

Configuring the Purge Service

To set configurations that will apply to the Purge service itself:

1. On the FTPC home page, click on the Purge service ID.
2. In the left panel, click **Service Configuration**. The Default Service Configuration screen displays.

Figure 11-7: Default Service Configuration Screen



The screenshot shows the 'Default Service Configurations' window. It is divided into several sections: 'Email' with fields for SMTP Hostname (smtp.datasweep.com), From Email Address (DataManagement), and Administrator Email; 'JMS' with fields for JMS URL (failover:tcp://localhost61616), User Name, User Password, and Purge Instance Name (3), plus a 'Test JMS' button; 'Running Parameters' with dropdowns for Connection Timeout (30 seconds), Database Query Timeout (10 minutes), Reaper Wait Time (None), Purge Records Per Query (10000), and Max Statement Per Batch (1000), and three checked checkboxes for 'Purge Unused Reference Designators', 'Purge Unused Instruction Lists', and 'Purge Unused Note Blobs'; 'Throttle' with a dropdown for SQL Throttle Suspend (5); and 'File Directory' with fields for Marker File Directory (C:\temp\purgeMarker) and Reaper File Directory (C:\temp\purgeReaper). At the bottom right are 'Cancel' and 'Save' buttons.

3. Define the following properties:

► **Email**

- ❖ *SMTP Hostname*: enter the name of the SMTP server. The SMTP Server manages all the notification email messages sent regarding Purge jobs. It manages the emails for the application server regardless of where a client machine is in the network. The application server contacts the SMTP server and sends the email.
- ❖ *From Email Address*: enter the default address to send notifications from. The default is DataManagement.
- ❖ *Administrator Email*: enter the site administrator's email address. This is the email address that will receive all notifications.

► JMS

- ❖ *JMS URL*: enter the connection information required to connect to the appropriate ActiveMQ broker in order to communicate with Purge. The format should be as follows:

failover: (<hostname>: <port>)

For example: **failover:(tcp://mqhost:61616)** Please note that the hostname must include the appropriate protocol.

IMPORTANT: Using a failover URL format ensures that if a connection is dropped (for example, due to inactivity), reconnection will be attempted.

The default ActiveMQ port is 61616. Purge and FTFC Administrator can use different ActiveMQ brokers.

Click [Test] to test the JMS connection. The connection must be running to perform the verification. However, the connection does not have to be running or verified to save the configuration.

- ❖ *JMS User*: enter the user name for a user who has privileges on the ActiveMQ broker. If a user name is not specified, the system will use ActiveMQ's default user name, which is *system*. Depending on your configuration, this property may be optional.
- ❖ *JMS Password*: enter the password corresponding to the user name defined in the **JMS User** property. If a password is not specified, the system will use ActiveMQ's default password, which is *manager*. Depending on your configuration, this property may be optional.
- ❖ *Purge Instance Name*: define the Purge instance name associated with this connection.

► Running Parameters

- ❖ *Connection Timeout*: select the timeout value for connecting to the database. If the Purge job cannot connect to the database by this time, the Purge job will be terminated. The default is 30 seconds.

- ❖ *Database Query Timeout*: select the timeout value for executing a SQL statement. If the SQL statement cannot be executed by this time, the Purge job will be terminated. The default is 10 minutes.
- ❖ *Reaper Wait Time*: select the time the reaper must wait after processing a purge record before processing another. This controls the load placed on the database server during a Purge job. The default is None.

Because the reaper issues DELETE statements continuously one after the other, its processing must be governed to control the load it puts on the database server. The load on the database server can be controlled by setting the reaper wait time.

- If you will be running Purge during peak hours, it would be helpful to set this property to a value that allows some time between issuing the DELETE statements to decrease the load on the database server.
- If you will be running purge at off-peak hours, then leaving this property at None is recommended because the load on the database server at that time is minimal, and leaving this property at None will help you get through the purge as quickly as possible.

- ❖ *Purge Records Per Query*: select the maximum number of purge records fetched by the reaper in one SQL statement. The default is 10000 records.
- ❖ *Max Statement Per Batch*: select the maximum number of statements in a batch before it is executed. Each batch is considered one execution. The default is 1000 statements.
- ❖ *Purge Unused Reference Designators*: select whether or not the deletion of reference designators that are not being used (i.e., not associated with any objects) is enabled. This feature is enabled by default.
- ❖ *Purge Unused Instruction Lists*: select whether or not the deletion of work instructions that are not being used (i.e., not associated with any objects) is enabled. This feature is enabled by default.
- ❖ *Purge Unused Note Blobs*: select whether or not the deletion of notes that are not being used (i.e., not associated with any objects) is enabled. This feature is enabled by default.

- ▶ **Throttle**: define the SQL Throttle Suspend value. This is the minimum time between the execution of SQL statements in milliseconds. This property controls the maximum number of queries issued to the database server from all the purge processes (marker and reaper). It does not affect any other Data Management or Plant Operations processes. Choose from none, 5 (default), 10, 25, and 50.
- ▶ **File Directory**
 - ❖ *Marker File Directory*: the directory where the file containing the purge records will be saved. This file is generated by the marker process. The default is C:\temp\purgeMarker.

- ❖ *Reaper File Directory*: the location of the file that is used by the reaper to identify the reaping candidates. The default is C:\temp\purgeReaper.
4. Click [Save] to save your configuration.

Sending Purge Configurations

Configuration files for Purge are written to `<FTPC_install>\FTPCAdminHome\adminData\<component>`. By default, this location is C:\Rockwell\PO<version>.<build>\FTPCAdminHome\adminData\<component>.

The following files store the Purge configuration:

- ❑ **DefaultServerJobConf.xml**: contains the default values for the all the properties of a Purge job. These values are applied to all Purge jobs unless the property is specifically defined in FTPC Administrator.
- ❑ **DSDataSource.xml**: contains the connection information to the databases to be purged.
- ❑ **DSServerJobConfig.xml**: contains the defined Purge job type, the site information for the databases to be purged, and any individual Purge job property that will use values other than the default values.
- ❑ **DSServerSchedule.xml**: contains the defined Purge schedule.

Configuration files for new or edited Purge jobs are stored in the **next** subdirectory, which is located at the same level as the **current** subdirectory and has the same subdirectory structure. The **current** subdirectory always reflects the Purge jobs' current configurations on their respective systems.

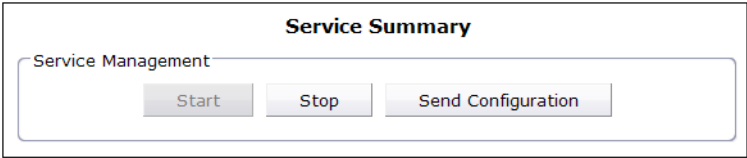
A new Purge service will only have property files in the **next** subdirectory. Once the configuration files are successfully sent to the target service, the **next** configurations are moved to the corresponding subdirectory in the **current** subdirectory and removed from the **next** subdirectory.

A modified configured Purge service will have property files in both its **current** and **next** subdirectories. Once the configuration files are successfully sent to the target service, the **next** configurations are moved to the corresponding **current** subdirectory and removed from the **next** subdirectory.

After you have created new configurations or updated existing ones, send the new and updated information to the Purge service.

1. Click the Purge service ID row in the Purge box on the home page. The following screen displays.

Figure 11-8: Service Summary Screen - Send Configuration



- 2. Click [Send Configuration]. This will move the configurations currently saved in the **next** subdirectory of the service to the corresponding **current** subdirectory, and the configurations in the **next** subdirectory are removed.

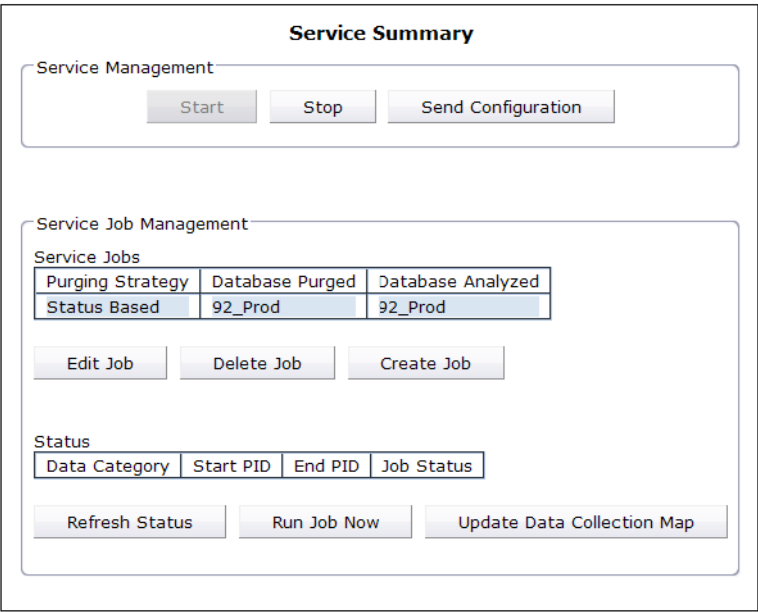
A dialog displays notifying you that the Purge service will be unavailable while the configuration is being sent.

Starting and Stopping Purge Jobs

To start a scheduled Purge job:

- 1. Click the Service ID row in the Purge box on the home page. The Service Summary screen displays.

Figure 11-9: Service Summary - Run Job



- 2. Click [Start] to start the Purge job. The job will run at the day and time scheduled.
- 3. Click [Stop] to stop a Purge job.

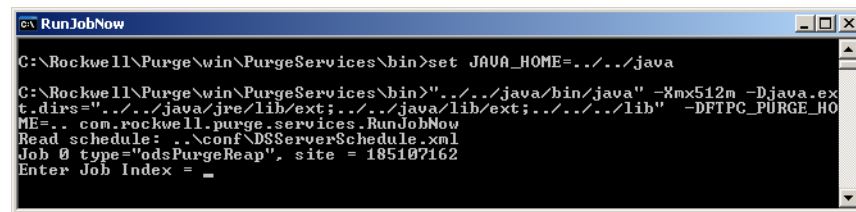
Running A Purge Job Manually

If you want to run a Purge job immediately, do one of the following:

- ☐ Click [Run Job Now] in the Services Summary screen.
- ☐ Go to Start > Programs > Rockwell Software > FTPC Purge <version> > Purge Services > RunJobNow. A command window displays listing the job index of the configured Purge jobs.

Enter the job index that corresponds to the job you want to run and press Enter to run the job.

Figure 11-10: Run Purge Job Now



```

C:\Rockwell\Purge\win\PurgeServices\bin>set JAVA_HOME=../../java
C:\Rockwell\Purge\win\PurgeServices\bin>"../../java/bin/java" -Xmx512m -Djava.ext
t.dirs=../../java/jre/lib/ext;../../java/lib/ext;../../lib" -DFTPC_PURGE_HO
ME=.. com.rockwell.purge.services.RunJobNow
Read schedule: ..\conf\DSServerSchedule.xml
Job 0 type="odsPurgeReap", site = 185107162
Enter Job Index = _
  
```

Viewing Purge Statuses

To view the status of a Purge job, click the Purge service ID row on the home page to display the Service Summary screen. The Purge status is displayed in the Service Job Management box.

Figure 11-11: Purge Status

Service Job Summary

Service Jobs:

Job Type	Database Purged	Database Analyzed
Status Based	a_jboss	h_jboss
Status Based	529C2C7968C14E19BD6DE557C9FD3D77	C2DE917C7AD4499CA5D857CF098E9ACC
Status Based	C2DE917C7AD4499CA5D857CF098E9ACC	
Status Based	529C2C7968C14E19BD6DE557C9FD3D77	

Job Type: odsPurgeResp, Site Number: 2

Data Category	Start Pid	End Pid	Job Status
TRX Record	0	0	N/A
Batches	0	0	N/A
ObjectStateHistory	0	0	N/A
ProcessOrders	0	0	N/A
Report and Labels	0	0	N/A
Runtime ActivitySets	0	0	N/A
Stateless Runtime Objects	0	0	N/A
Tracked Runtime Objects	0	0	N/A
Transactions	0	0	N/A
Workflow Objects	0	0	N/A

To update the status, click [Refresh Status].

Updating Data Collection Maps (Status-Based Only)

It is important to regularly update the Data Collection Set (DCS) maps so that objects that need to be purged will be marked correctly. After you have run a Status-Based Purge for the first time, your DCS maps will be automatically updated

NOTE: We recommend you update your DCS maps before each Purge in the future.

To update the DCS maps manually when Purge is not running, do one of the following:

- ❑ In FTPC Administrator, click the Purge service ID row on the home page and then click [Update Data Collection Map] in the Service Summary screen (Figure 11-9).
- ❑ Perform one of the following depending on your environment:
 - ▶ **Windows:** Go to Start > Programs > Rockwell Software > FTPC Purge <version_number>. <build_number> > RunDCSMapUpdate
Enter the Site Num and the Site GUID for the active database when prompted.

TIP: These values can be found in the SITE_INFO table for your active database.

- ▶ **Linux:** Change to the <LT_install>/linux/PurgeServices/bin/ directory and run the *RunDCSMapUpdate.sh* file.

The discovered relationships are stored in the PURGE_DCS_MAP table. Verify the DCS relationships in this table.

See “DCS Relationships” on page 43 for more information.

Uninstalling and Upgrading Data Management

In this chapter

- ❑ **Uninstalling Live Transfer 188**
 - Windows 188
 - Linux 189
- ❑ **Upgrading Live Transfer 190**
 - Windows 190
 - Linux 190
- ❑ **Uninstalling Purge 190**
 - Windows 191
 - Linux 192
- ❑ **Upgrading Purge 192**
 - Additional Default Properties 192
 - Windows 193
 - Linux 193

Uninstalling Live Transfer

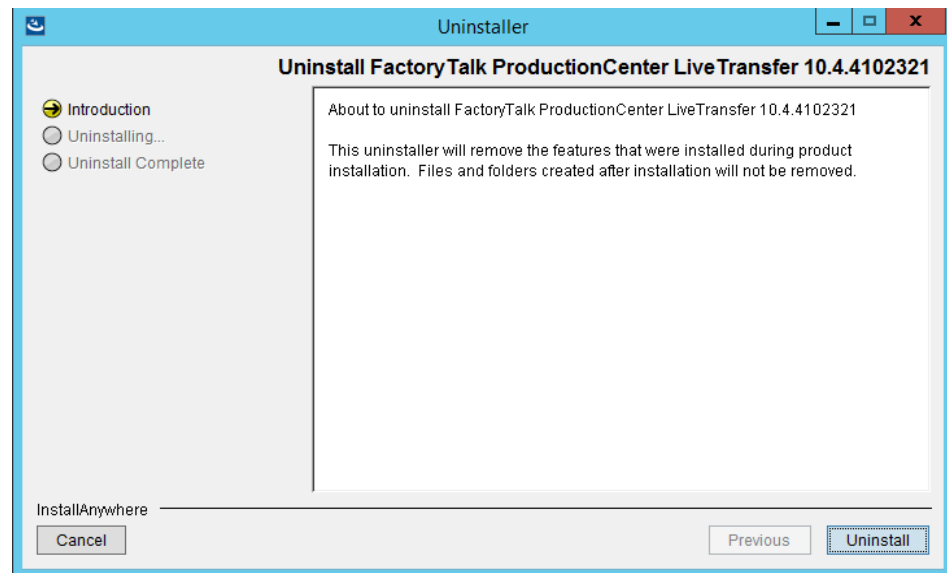
Perform the following steps to uninstall Live Transfer depending on your environment.

Windows

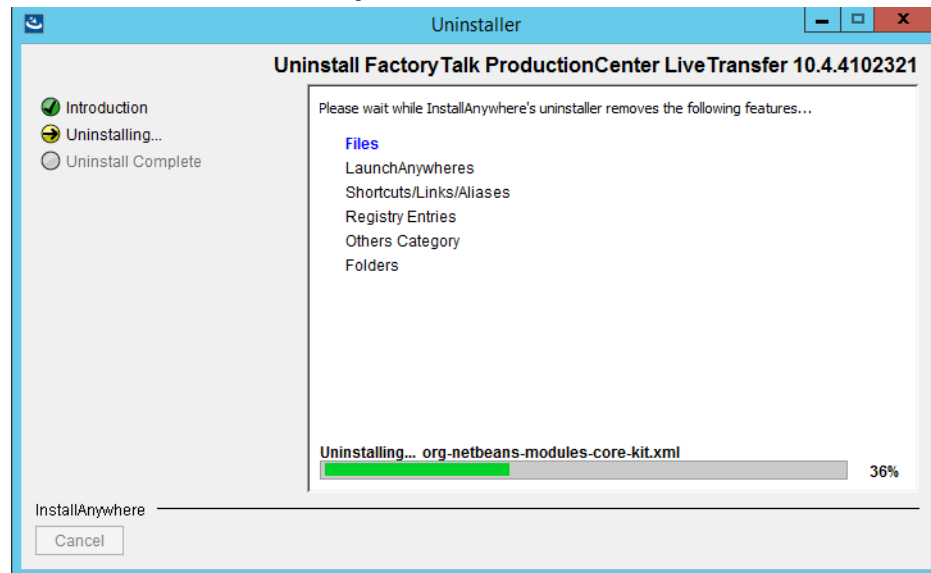
To uninstall Live Transfer:

1. Stop and unregister ActiveMQ and the Live Transfer components. See “Stopping and Unregistering Live Transfer Components” on page 76.
2. Go to `<LT_install>\uninstall` where `<LT_install>` is the location where you installed Live Transfer.
3. Double-click *Uninstaller.exe*. The uninstall welcome screen displays.

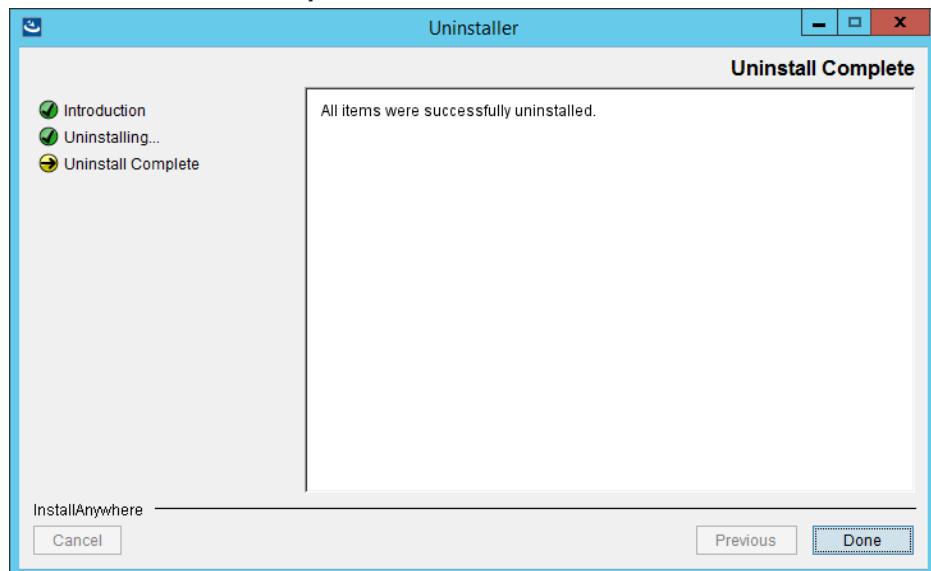
Figure 12-1: Uninstall Welcome Screen



4. Click [Uninstall]. The installer displays a progress screen as Live Transfer is uninstalling.

Figure 12-2: Uninstall Summary Screen

The Uninstall Complete screen displays when the uninstall is complete.

Figure 12-3: Uninstall Complete Screen

5. Click [Done] to exit the uninstaller.

Linux

To uninstall Live Transfer, stop the Live Transfer service and then delete the installation directory on your Linux machine.

Upgrading Live Transfer

Perform the following steps to upgrade Live Transfer depending on your environment.

Windows

To upgrade Live Transfer:

1. If you want to use the same configuration, copy the *SiteExtractManager.properties*, *TableExtractManager.properties*, and *TableLoaderManager.properties* files and any other configuration file that you may have changed (for example, *logging.properties*) into another location.
2. Uninstall the old version of Live Transfer. See “Uninstalling Live Transfer” on page 188 for details.
3. Download the new Live Transfer file from the Rockwell Software site.
4. Install the new version of Live Transfer. See “Installing Live Transfer” on page 66 for details.
5. Copy the configuration files that you saved in Step 1 into the appropriate folders.

Linux

To upgrade Live Transfer:

1. If you want to use the same configuration, copy the *SiteExtractManager.properties*, *TableExtractManager.properties*, and *TableLoaderManager.properties* files and any other configuration file that you may have changed (for example, *logging.properties*) into another location.
2. Uninstall the old version of Live Transfer. See “Uninstalling Live Transfer” on page 188 for details.
3. Download the new Live Transfer file from the Rockwell Software site.
4. Install the new version of Live Transfer. See “Installing Live Transfer” on page 66 for details.
5. Configure the installation on your Linux machine. See “Installing Live Transfer on Linux” on page 69 for details.
6. Copy the configuration files that you saved in Step 1 into the appropriate folders.

Uninstalling Purge

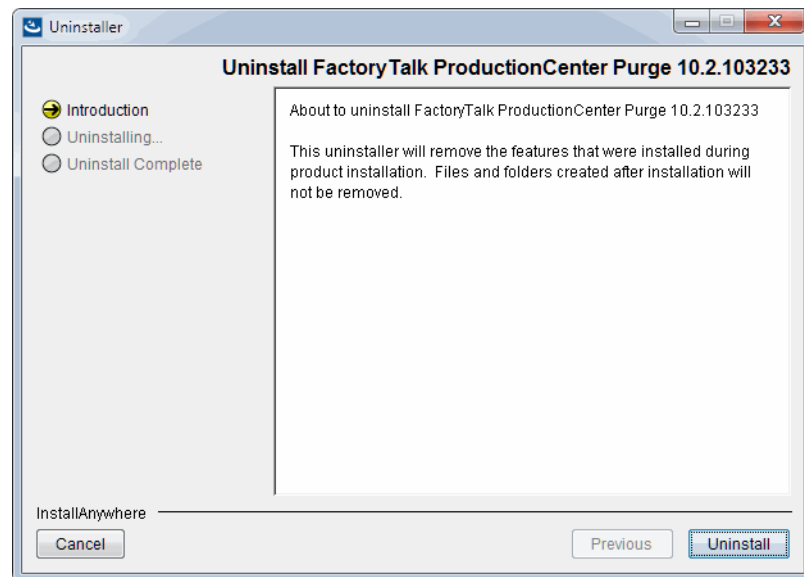
Perform the following steps to uninstall Purge depending on your environment.

Windows

To uninstall Purge:

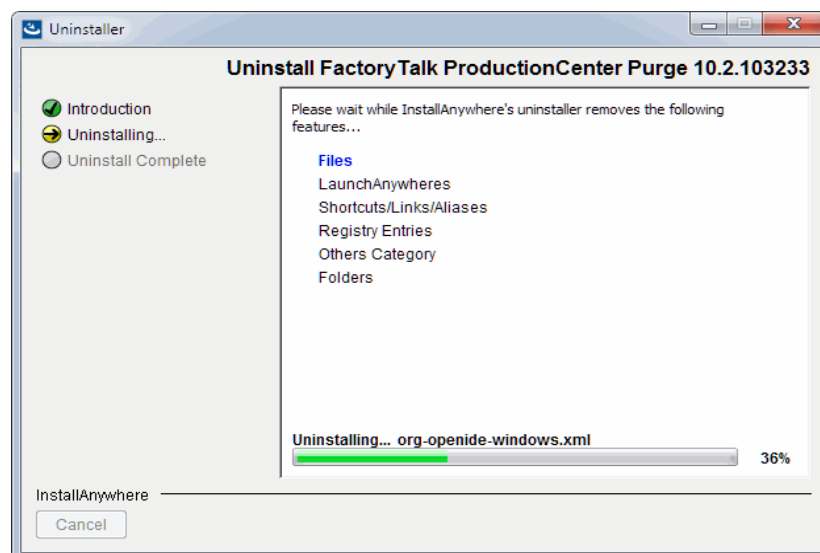
1. Stop and unregister ActiveMQ and Purge. See “Stopping and Unregistering the Purge Service” on page 88 for details.
2. Go to `<Purge_install>\uninstall` where `<Purge_install>` is the location where you installed Purge.
3. Double-click *Uninstaller.exe*. The uninstall welcome screen displays.

Figure 12-4: Uninstall Welcome Screen



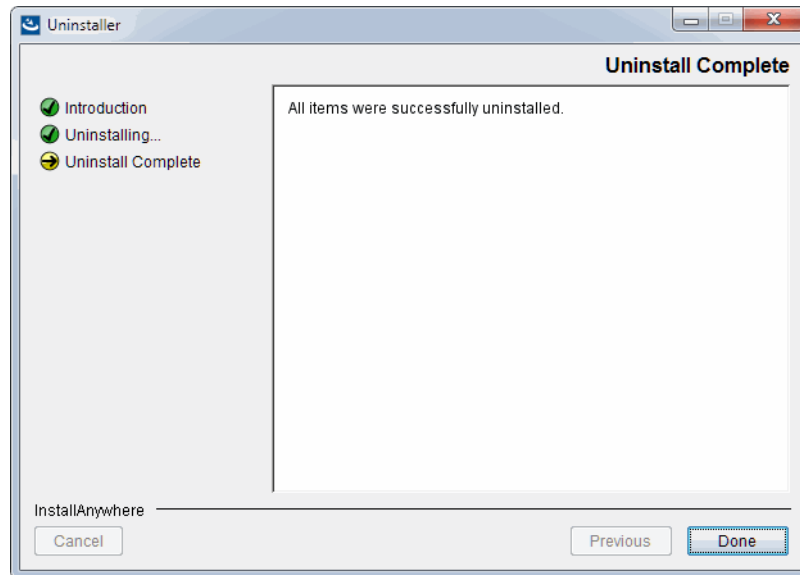
4. Click [Uninstall]. The installer displays a progress screen as Purge is uninstalling.

Figure 12-5: Uninstall Summary Screen



The Uninstall Complete screen displays when the uninstall is complete.

Figure 12-6: Uninstall Complete Screen



5. Click [Done] to exit the uninstaller.

Linux

To uninstall Purge, stop the Purge service and then delete the installation directory on your Linux machine.

Upgrading Purge

Perform the following steps to upgrade Purge depending on your environment.

Additional Default Properties

If you are upgrading from a pre-9.3.103193 Purge build to a 10.4 Purge build, either use the new DefaultServerJobConf.xml file provided with your upgrade or make sure these two properties have been added to your existing DefaultServerJobConf.xml file as follows.

```
<items>
  <item-name>PurgeConnectRetry</item-name>
  <item-value>10</item-value>
</items>
```

and

```
<items>
  <item-name>ODSMarkerCandidateSelectRowLimit</item-name>
  <item-value>5000</item-value>
</items>
```

The properties are defined as follows:

- ❑ **PurgeConnectRetry**: the number of times the Purge process will attempt to connect to the database before an error is thrown. The default is 10.
- ❑ **ODSMarkerCandidateSelectRowLimit**: the number of rows fetched at one time when the rows are being marked by the ODS marker. The marker goes through the entire returned record set but in a batch size defined by this parameter. The default is 5000.

If you are upgrading from a pre-10.2.103285 Purge build, either use the new DefaultServerJobConf.xml file provided with your upgrade or make sure the following property has been added to your existing DefaultServerJobConf.xml file as follows.

```
<items>
  <item-name>PurgeConnectionValidate</item-name>
  <item-value>true</item-value>
</items>
```

This property automatically validates the Purge connection when it is started. The default is *true*.

Windows

To upgrade Purge:

1. If you want to use the same configuration, copy the *jms-local.config*, *DSDatasource.xml*, *DefaultServerJobConf.xml*, *DSServerSchedule.xml*, and *DSServerJobConfig.xml* files into another location.
2. Uninstall the old version of Purge. See “Uninstalling Purge” on page 190 for details.
3. Download the new Purge file from the Rockwell Software site
4. Install the new version of Purge. See “Installing Purge” on page 80 for details.
5. Copy the configuration files that you saved in Step 1 into the appropriate folders.

Linux

To upgrade Purge:

1. If you want to use the same configuration, copy the *jms-local.config*, *DSDatasource.xml*, *DefaultServerJobConf.xml*, *DSServerSchedule.xml*, and *DSServerJobConfig.xml* files into another location.
2. Uninstall the old version of Purge. See “Uninstalling Purge” on page 190 for details.
3. Download the new Purge file from the Rockwell Software site.
4. Install the new version of Purge. See “Installing Purge” on page 80 for details.
5. Configure the installation on your Linux machine. See “Installing Purge on Linux” on page 83 for details.
6. Copy the configuration files that you saved in Step 1 into the appropriate folders.

Troubleshooting

In this chapter

- ❑ **Understanding Live Transfer Messages** 196
- ❑ **Recording Errors** 197
- ❑ **Handling a Communication Loss** 197
- ❑ **Encrypting Data** 198

The following sections describe troubleshooting methods for Live Transfer and Purge issues you may encounter.

Understanding Live Transfer Messages

This section explains some of the more complex messages that you might see in the SiteExtractManager log when running Live Transfer. This log is located in the following locations:

- ❑ **Windows:** <LT_install>\win\LiveTransferExtractor\logs
- ❑ **Linux:** <LT_install>/linux/LiveTransferExtractor/logs

Message: INFO: Sending extracted data to loader. Table=TRX_BASE, pidRange=[173, 176), extractNumber=1

Meaning: This message is generated when a table extractor has performed a query and has determined that data actually needs to be transmitted to the loader. In other words, records were found that have either been updated since they were first transmitted or new un-transmitted records were found by the table extractor.

The *extractNumber=1* parameter indicates which set of extract rows has been sent. If the result of the extract query produces a large number of rows that will need to be extracted in sets, this counter shows which message of the set has been sent.

Message: WARNING: Too many unprocessed settled PIDs outstanding (count=6). Blocking ExtractTableRequests until count is reduced.

Meaning: This message occurs when the loader is not keeping up with the extractor.

Specifically, a number of PIDs that are old enough to be committed have been completely extracted and sent to the loader, but the confirmations for all of the load requests have not been received. As a result, the SEM is going to stop issuing new extract requests. The TLM will continue loading everything in its queue, including any PIDs that had been missed. Once the SEM receives the loader confirmation for the PIDs that are missing confirmations, it will start extracting again.

Message: INFO: Resend Extract Request: ExtractRequest: table=RT_ACTIVITY_SET_BLOB, [2, 5), resend=true, allSettledPids=true

Meaning: This message indicates that the **Resend Timeout** value (configured in “Defining the Source Database” on page 146) has been exceeded for a specific load request. The extractor is ending the current extraction and requesting a re-extraction.

Message: INFO: No receipt found for table RT_ACTIVITY_SET. This table will be re-extracted.

Meaning: This message occurs when the SEM has stopped extracting from a specific table and has issued a re-extraction, but the loader had actually been processing the load request. This message is generated when the acknowledgement of the load request is received by the SEM, but there is no longer a receipt waiting to be acknowledged.

Message: INFO: Delaying Posting Extract Table Requests.

Meaning: This message means that the table extract requests are not being processed within the configured extract period.

In order to try and prevent out-of-order message processing by the loader, the SEM will check the number of messages on the queue from which the table extractor instances are reading the extract table requests prior to posting new table extract requests. If there are too many un-processed messages, the SEM will delay posting new requests to prevent sending data to the loader for a later PID range prior to receiving load data for the previous PID range.

Message: INFO: UnacknowledgedRequests count=42, max=207

Meaning: This message is generated by the SEM and gives a report on the number of unacknowledged load requests currently in the receipt cache. The build-up of unacknowledged receipts is the first indication to the SEM that the TLM instance is no longer available.

Recording Errors

All errors are written to the error log files. See “[Viewing Live Transfer Logs](#)” on [page 77](#) for details.

Handling a Communication Loss

If the connection to Live Transfer or Purge is lost while a job is running, the job will pick up where it left off when the connection is re-established.

In the case of Live Transfer’s extract service, if you have multiple TEMs and one of them becomes unavailable, the extract work will be performed by the remaining TEM instances. In the case of the load service, if you have multiple TLMs and one of them becomes unavailable, the load work will be performed by the remaining TLM instances.

If data-content messages are not acknowledged in a timely manner, the extractor service will re-send the messages.

Encrypting Data

Connections to the JMS providers include authentication at the username/password level. If you want to encrypt the data being transferred, the deployment can use ActiveMQ's SSL and/or HTTPS support to encrypt the data. Please see your ActiveMQ documentation for more information.

Index

A

- Access Control 132
- ActiveMQ 34, 139
 - configuring 160
 - connection 138
 - documentation 16
 - firewall environment 160
 - ports 69, 83
 - recommendations 27, 166
 - registering 73, 86
 - separate cluster 160
 - starting 73, 74, 86, 87
 - stopping 77, 88
 - unregistering 77, 88
- Administrator Log 124
- ALL Mode 22, 148
- Application Log 124
- Association Rules
 - batches 46, 57
 - process orders 48
 - tracked runtime objects 55
- ATRows 58
- Audit Tables 117
 - runtime activity sets 123

B

- Base ODS 16, 100
 - migrating 108
- Basic Rules
 - batches 45, 56
 - process orders 48
- Batches 41, 45
- Binary Large Object (BLOB) 71
- BOM

- object uniqueness 130

C

- Client Inactivity Timeout 132
- Closing Consumed Lots 131
- Communication Loss 197
- Comparison Tools
 - LTPIDCompare 156
 - LTSampleCompare 157
- Configuring
 - access control 132
 - ActiveMQ 160
 - audit tables 121
 - database logging 119
 - email services 141
 - filtering 139
 - Live Transfer 146
 - logging 159
 - PID increment interval 133
 - Purge 173
 - serial numbers 141
 - sublot naming 140
 - sublot quantity constraint 134
 - user login requirements 142
- Consolidated Log 40
- Consumed Parts Consolidation 131
- Consumption Rules
 - batches 46, 57
 - tracked runtime objects 54
- Containment Rules
 - batches 46, 57

D

- Data
 - deleting runtime data 144

- deleting work orders 144
- non-static 29
- static 29
- Database
 - audit tables 117, 121
 - BODS 100, 108
 - history logging 119
 - HODS 100
 - initializing 106
 - logging 119
 - migrating 14, 107
 - object revisioning 116
 - ODS 14, 100
 - Oracle 103
 - Oracle RAC 103
 - pd_xfr_update_pid 108
 - primary tables 117
 - Production 100
 - registering 101
 - remapping 110
 - reorganizing 110
 - reporting database 14
 - schema versions 14
 - site number 129
 - SQL 103
 - SQL Mirroring 103
 - transaction logging 118
 - upgrading 109
 - user login 10
- Database Logging
 - administrator log 124
 - application log 124
- DCS Relationships 43
- DefaultServerJobConf.xml 193, 194
- Deleting
 - runtime data 144
 - work orders 144
- Distributed Events 134, 136
 - keyed object changes 138
- Documentation
 - ActiveMQ 16
 - FactoryTalk ProductionCentre 9
 - third-party 9
- DSDatasource.xml 193, 194
- DSServerJobConfig.xml 193, 194
- E**
 - Email Services 141
 - Extract Queue 19

- F**
 - FactoryTalk ProductionCentre
 - documentation 9
 - Filtering 139
 - Flexible State Model (FSM) 45, 48, 56
- G**
 - Group Rules
 - batches 46, 56
 - process orders 48
 - tracked runtime objects 52
 - GUID 34
- H**
 - Hardware Requirements 9
 - History Logging 119
 - Hub ODS 16, 100
- I**
 - Initializing Database 106
 - Installing
 - Live Transfer 66
 - Live Transfer on Linux 69
 - Purge 80
 - Purge on Linux 83
 - Inventory Container Histories 43
- J**
 - Java Service Wrapper 164
 - Java Virtual Machines (JVMs) 18
 - JBossCache 16
 - JGroups 139
 - JMS
 - connection 138
 - jms-local.config 193, 194
- L**
 - Labels 40, 48
 - LIST Mode 23, 148
 - Live Transfer
 - communication loss 197
 - components 16
 - configuring 146
 - deployment options 165
 - encrypting data 198
 - error recording 197
 - installing 66
 - installing on Linux 69
 - logs 77

- overview 18
- recommendations 165
- setup 15
- stopping 76
- uninstalling 188
- unregistering 76
- upgrading 190
- Live Transfer Extractor 16
 - starting 74
- Live Transfer Extractor (LTE)
 - recommendations 165
 - registering 73
 - starting 74
- Loader Queue 17, 20
- Logging 159
- logging.properties 159
- Lots 54
 - closing consumed lots 131
 - consumed parts consolidation 131
- LT_EXTRACT_TABLE_LOG 23

M

- Marker Process 30
 - aborting 32
- Marker Request Queues 35
- Message Packs 143
- Messaging
 - ActiveMQ 139
 - JGroups 139
- Migrating Databases 14, 107

O

- Object Changes
 - event distribution 138
- Object Revisioning 116
- Object Uniqueness 130
- ObjectStateHistories 42, 47
- ODS Database 100
- ODS Marker 30
- Oracle
 - database 103
 - RAC 103

P

- Pallets 41
- Parts
 - object uniqueness 130
- pd_xfr_update_pid 108
- PID Increment Interval 133
- PIDs 19, 31, 36, 37

- populatePdXfrUpdatePid.bat 14, 73
- Primary Tables 117
- Process Orders 41, 47
- Production Database 100
- Production Marker 30
- Purge
 - application tables 29
 - ATRows/tracked object dependency 58
 - communication loss 197
 - configuring 173
 - installing 80
 - installing on Linux 83
 - instance tables 28
 - limitations 27, 35
 - logs 89
 - overview 33
 - processes 29
 - reaper 32
 - registering 73, 86
 - SQL throttle 63
 - starting 86
 - stopping 88
 - uninstalling 190
 - unregistering 88
 - upgrading 192
- Purge Request Queue 34
- Purge Service 34
- Purge Subsystem 34
- PURGE_RECORD 30, 32
- PurgeListener 34

R

- Reaper Process 32
 - aborting 32
- Receipt Object 17, 20
- Receipt Queue 20
- Registering
 - ActiveMQ 73, 86
 - databases 101
 - Live Transfer Extractor (LTE) 73
 - Purge 73, 86
 - Table Loader Manager (TLM) 73
- Remapping Database 110
- Reorganizing Database 110
- Reporting Database 14
- Reports 40, 48
- Runtime Activity Sets
 - auditing 123

S

- Scheduler 34
- Schema Versions 14
- Semantic Properties
 - sys.allow.Batch.purge 45, 56
 - sys.allow.ProcessOrder.purge 48
- Serial Numbers 141
- Shop Operations
 - client inactivity timeout 132
- Site Extract Manager (SEM) 16, 17
 - extract re-processing 22
- Site Number 129
- SiteExtractManager.properties 190
- SOAP Protocol 140
- Source Database
 - schema version 14
- SQL
 - database 103
 - mirroring 103
 - throttle 63
- Starting
 - ActiveMQ 73, 74, 86, 87
 - Live Transfer Extractor 74
 - Live Transfer Extractor (LTE) 74
 - Purge 86
 - Table Loader Manager (TLM) 74
- State Change Events
 - distributed events (Client) 134
 - distributed events (Server) 136
- State Change Events (Client) 134
- State Change Events (Server) 136
- Stateless Runtime Objects 40, 49
- Status-Based Purge 39
 - batches 41, 45
 - consolidated log 40
 - criteria 39
 - DCS relationships 43
 - inventory container history 43
 - labels 40, 48
 - ObjectStateHistories 42, 47
 - ODS Marker 30
 - pallets 41
 - process orders 41, 47
 - Production Marker 30
 - reports 40, 48
 - stateless runtime objects 40, 49
 - tracked objects 42
 - tracked runtime objects 51
 - transactions 40, 55

- workflows 40, 55

Stopping

- ActiveMQ 77, 88
- Live Transfer 76
- Purge 88

- Sublot Naming 140

- Sublot Quantity Constraint 134

- SublotRelation 45

- Sublots 45, 56

- System Messages

- updating 143

T

- Table Extract Manager (TEM) 16, 17
 - Manager 17

- Table Extractor instances 17

- Table Loader Manager (TLM) 16, 18
 - Manager 18

- recommendations 166

- registering 73

- starting 74

- Table Loader instances 18

- TableExtractManager.properties 190

- TableLoaderManager.properties 190

- Target Database

- schema version 14

- Time-Based Purge 36

- criteria 36

- marker 30

- Tracked Objects 42, 58

- Tracked Runtime Objects 51

- Transaction Logging 118

- Transactions 40, 55

U

- Uninstalling

- Live Transfer 188

- Purge 190

- Unregistering

- ActiveMQ 77, 88

- Live Transfer 76

- Purge 88

- Updating

- system messages 143

- Upgrading

- Live Transfer 190

- Purge 192

- Upgrading Databases 109

- User Authentication 140

- User Groups 133

User Login 142

V

Version Compatibility 14

W

Workflows 40, 55

X

XFR_EXTRACT_LOG 22

XFR_LOAD_LOG 22

