Upgrading Software AG Products

Version 9.9

October 2015

This document applies to webMethods Product Suite Version 9.9 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: WEBM-UPG-99-20151015

Table of Contents

About this Guide

This guide explains how to upgrade Software AG products, specifically Apama, Terracotta, and webMethods.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention *folder.subfolder:service* .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
Italic	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
`Monospace font`	Identifies text you must type or messages displayed by the system.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at https://empower.softwareag.com.

To submit feature/enhancement requests, get information about product availability, and download products, go to Products.

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the Knowledge Center.

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at http://techcommunity.softwareag.com. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, code samples, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

1 Upgrades Covered in This Guide

Upgrades Covered in This Guide

This guide explains how to upgrade Apama, Terracotta, and webMethods products from release 9.0, 9.5, 9.6, 9.7, or 9.8 to release 9.9 using a side-by-side procedure.

Note: The information provided in this guide for Terracotta Server Array and Universal Messaging applies only to when you are using those products with webMethods products.

Upgrades Not Covered in This Guide

The upgrades listed below are not covered in this guide. Instead, see the indicated information source.

Upgrade	Information
From a release prior to 9.0	See *Supported Upgrade Paths*
ApplinX	See ApplinX documentation
webMethods Broker to Universal Messaging	See *Migrating from webMethods Broker to webMethods Universal Messaging*
Content Service Platform	Contact Software AG Global Consulting Services
MashZone or Process Performance Manager	See MashZone or Process Performance Manager documentation
Mobile Administrator on a virtual machine (This guide explains how to upgrade Mobile Administrator on a physical machine)	Uninstall Mobile Administrator from the virtual machine (see *Installing Software AG Products old_release*) and install Mobile Administrator 9.9 (see *Installing Software AG Products* 9.9)
Presto releases prior to 3.6	Contact Software AG Global Consulting Services
Terracotta Server Array when not deployed in support of webMethods products	See Terracotta BigMemory Max documentation

Upgrade	Information
Universal Messaging when not used with webMethods products	See Universal Messaging documentation

Release Numbering Exceptions

The products below did not follow general release numbering.

Product	Release Numbering
ActiveTransfer	In the 9.9 release, the ActiveTransfer release number is 9.8, because the ActiveTransfer release is the same release that was available with the 9.8 release.
webMethods Broker	In the 9.6, 9.7, 9.8, and 9.9 releases, the webMethods Broker release number is 9.6, because the webMethods Broker release is the same release that was available with the 9.6 release.
Presto	In the 9.6, 9.7, and 9.8 releases, the Presto release numbers were 3.7, 3.8, and 3.9, respectively. The Presto release before Presto became part of webMethods was 3.6.
Terracotta	In the 9.0, 9.5, 9.6, 9.7, 9.8, and 9.9 releases, the Terracotta release numbers were 3.7.4, 3.7.6, 3.7.6, 4.1.4, 4.3, and 4.3.1, respectively.

Guides Needed to Perform Upgrades

You will always need the guides listed below to perform the upgrades in this guide.

- *Using the Software AG Installer* (October 2015 release)
- *Installing Software AG Products* 9.9
- *Using the Software AG Update Manager*

Depending on the products you are upgrading, you might need the 9.9 guides and other guides listed below to perform the upgrades in this guide.

- *webMethods Application Platform User's Guide*
- *webMethods BPM Task Development Help*
- *Using Blaze Rules with BPM and CAF*
- CentraSite documentation

- *webMethods Deployer User's Guide*
- *webMethods Integration Server Administrator's Guide, webMethods Service Development Help,* and *webMethods Integration Server Clustering Guide*
- *Administering webMethods OneData*
- *Administering webMethods Optimize* and *Configuring BAM*
- Presto documentation
- *Getting Started with the webMethods Product Suite and Terracotta* and Terracotta BigMemory Max documentation

2 Critical Requirements and Recommendations For Successful Upgrade

Critical Upgrade Requirements

This chapter describes critical requirements that must be followed to make your upgrade successful.

Important: If you do not conform to these requirements, you will experience unpredictable results, possibly including corruption of your installation and data.

- Read the 9.9 product readmes, including the 9.9 readme for the Software AG Infrastructure, available on the Software AG Documentation website. The product readmes contain this information:
  - Critical information, known issues, fixes, resolved issues that had not been rolled into fixes at the time of the release, and usage notes for your products.
  - Changes to product behavior, services, parameters, properties, APIs, documentation, and terminology. This information is especially important because you might need to modify product files or assets after migration to accommodate the changes.

  The 9.9 readmes contain information for release 9.5 and later; read the information for your old release + 1 through the 9.9 release. For example, if you are upgrading from release 9.0, read the information for releases 9.5 through 9.9.
- Read the product release notes; the release notes provide information on new functionality. Read the information for every release for your old release + 1 through the 9.9 release.
- Perform the tasks in this guide in the exact order in which the tasks are presented. The task order is critical because your products have many inter-dependencies, including a shared infrastructure.
- Do not start any 9.9 products before the instructions in this guide tell you to do so, or your database components could become corrupted.
- Perform the tasks in this guide for all upgrade paths unless the guide states that a particular task is required for certain upgrade paths only.
- In the 9.7, 9.8, and 9.9 releases, the webMethods Broker release number is still 9.6 and is the same release that was available with the 9.6 release. Even though the release is the same, you must still perform the tasks described in this chapter and this guide for webMethods Broker.

Recommendations and Tested Scenarios

- Software AG strongly recommends that you upgrade in a controlled test environment and test that environment for proper operation before upgrading your development or production environment.
- Software AG tests migration across machines that have the same hardware and operating systems. Integration Server also tests migration across machines that have different hardware or different operating systems. However, for other products, migration across machines that have different hardware or operating systems might have unpredictable results, especially when one machine is Windows and the other is non-Windows.
- This guide assumes you are using the same RDBMS in the new environment that you used in the old environment. If you want to use a different RDBMS in the new environment, there are special requirements that must be met before you can upgrade your products. Contact Software AG Global Consulting Services for more information.

API-Portal Requirements

You must have these functional privileges to upgrade API-Portal:

- License, user, and document management
- User management configuration
- Database administration

Ask your API-Portal administrator to log on to the User Management Component at http://*host* :*port* /umc, go to the user management page, click your user name, and either add you to the API Administrator group or go to the Functional Privileges tab and assign the privileges listed above to you.

9.6, 9.7, or 9.8 Upgrade: webMethods Broker Requirements

In the 9.7, 9.8, and 9.9 releases, the webMethods Broker release number is still 9.6 and is the same release that was available with the 9.6 release. If the existing webMethods Broker 9.6 from the 9.6, 9.7, or 9.8 release is installed on its own machine, you can continue using it with your 9.9 products. However, you must still follow all instructions and perform all tasks for webMethods Broker in this chapter and in "Complete the webMethods Broker Upgrade" on page 46.

If you want to upgrade to a machine that is running a different operating system or has different hardware than the old machine, or if you want webMethods Broker to exist with other 9.9 products in the same installation directory, you must install webMethods Broker 9.6 from the 9.9 release. You can install on a machine that is running the same or higher operating system and has the same hardware (for example, both have AMD, or both have Intel) as the old machine. Make sure you follow all instructions and perform all tasks for webMethods Broker in this chapter and in "Complete the webMethods Broker Upgrade" on page 46.

CentraSite Requirements

If you want to install the CentraSite Application Server Tier and CentraSite Registry Repository on different machines, or in the same directory but at different times, you must install the Registry Repository first.

If you install CentraSite 9.9 on the same machine as the old CentraSite, you will not be able to run both products at the same time, even if they use different ports, because of conflicts for IPC and shared memory resources.

Integration Server Clustering Requirements

In Integration Server, a cluster is defined as multiple nodes that point to the same ISInternal database component. Nodes can be Integration Server installations on different machines or server instances within an Integration Server installation.

1. Go to "Install 9.9 Products, Prepare Old Environment, Shut Down Old Products, Migrate Database Components, Prepare to Migrate Data, and Upgrade the Software AG Infrastructure" on page 23 and use the instructions to perform the steps below.

    a. Install a cluster of 9.9 Integration Servers that matches your old cluster. You can install the 9.9 Integration Servers in parallel.

       The installation instructions include installing fixes. You can install the fixes in parallel.

    b. Perform the rest of the tasks in the chapter. When shutting down the old products, shut down all cluster nodes.

2. Go to "Complete the Integration Server and Hosted Wm Packages Upgrade" on page 55 and use the instructions to perform the steps below.

    a. Read the sections at the beginning of the chapter and then run the Integration Server migration utility to migrate server instances from one old Integration Server installation to one Integration Server 9.9 installation. The utility will ask separately about each server instance. The first migration of a server instance must be run alone (that is, not in parallel).

       Then run the utility for the other old and 9.9 Integration Server installations. You can run the utility on the other installations in parallel, because migrations after the first migration only change the local file system.

    b. For each server instance, perform the rest of the tasks in the chapter. Make sure the 9.9 server instances are processing client requests.

> **Important:** If the old server instances were not configured to use a Terracotta-based clustering solution, configure the 9.9 server instances to use a Terracotta-based clustering solution. Support for Coherence was discontinued in release 9.0. For instructions, see *webMethods Integration Server Clustering Guide* 9.9.

3. Perform the rest of the steps in this guide, including "Complete Final Upgrade Tasks for All Products, and Troubleshooting" on page 115. That chapter includes a step to configure your products. For instructions on configuring Integration Server clusters, see the *webMethods Integration Server Clustering Guide*.

Mobile Administrator Requirements

Mobile Administrator 9.9 requires a different license from previous releases. Obtain a Mobile Administrator 9.9 license file from Software AG.

My webMethods Server Requirements

Environment Variable

In your old release, you might have identified a default My webMethods Server installation in the global path of the installation’s host machine using the environment variable WM_HOME. If you are going to install the new release on the same machine as the old release, the existence of this variable could cause problems, so you must remove the environment variable from the global path.

My webMethods Server Clustering Requirements

In My webMethods Server, a cluster is defined as multiple nodes that point to the same MywebMethodsServer database component. Nodes can be My webMethods Server installations on different machines or server instances within a My webMethods Server installation.

1. Go to "Install 9.9 Products, Prepare Old Environment, Shut Down Old Products, Migrate Database Components, Prepare to Migrate Data, and Upgrade the Software AG Infrastructure" on page 23 and use the instructions to perform the steps below.

    a. Install a cluster of My webMethods Server 9.9 installations that matches your old cluster. You can install the 9.9 My webMethods Servers in parallel.

       The installation instructions include installing fixes. You can install the fixes in parallel.

   b. Perform the rest of the tasks in the chapter except you do not need to upgrade the Software AG Infrastructure. When shutting down the old products, shut down all cluster nodes.

2. Go to "Complete the My webMethods Server Upgrade" on page 49 and follow the instructions to perform the steps below.

   a. Read the sections at the beginning of the chapter and then run the My webMethods Server migration utility to migrate server instances from one old My webMethods Server installation to one My webMethods Server 9.9 installation. The utility will ask separately about each server instance. The first migration of a server instance must be run alone (that is, not in parallel) because the first migration will make changes to the MywebMethodsServer database component.

      Then run the utility on the other old and 9.9 My webMethods Server installations. You can run the utility on the other installations in parallel, because migrations after the first migration only change the local file system.

   b. For each My webMethods Server installation, migrate the My webMethods Server infrastructure. You can migrate the My webMethods Server infrastructure on all My webMethods Server installations in parallel.

   c. Initialize one cluster node. Then initialize the other cluster nodes, but do not update the host names again. You can initialize the other cluster nodes in parallel.

3. Perform the rest of the steps in this guide, including "Complete Final Upgrade Tasks for All Products, and Troubleshooting" on page 115. That chapter includes a step to configure your products. For instructions on configuring My webMethods Server clusters, see *Administering My webMethods Server*.

Prepare the Databases

1. Check your RDBMS against *System Requirements for Software AG Products* 9.9. If the RDBMS version you are using is not supported by your 9.9 products, upgrade to a supported RDBMS version.

2. Some products, like Integration Server, Optimize, Process Engine, and Trading Networks, offer features to archive or purge data from their database components. You can reduce the amount of time needed to migrate database components later in this procedure if you archive and purge them now. For instructions, see the product documentation.

3. Back up the old databases.

   If you are upgrading My webMethods Server, back up the My webMethods Server installation directory at the same time you back up the database. If you have problems, you will need to restore data from both types of backup.

4. Software AG recommends using cloned databases when testing your upgrade. Cloning operations are typically done at the level of the schema (Oracle) or database (SQL Server and DB2) that contains the product database components, or at the level of the database user that owns all the database components. Data cloning is usually performed using export and import tools that are bundled with the database. For cloning procedures, see your database vendor documentation.

   Software AG recommends the following:

   - *Installing Software AG Products* describes the basic grants and privileges needed to work with product database components. Before cloning the databases, give the database users that will work with the cloned databases the same basic grants and privileges that were given to the database users that work with the live databases.
   - In some cases, one database user grants permissions to a second user (for example, the process audit database user grants permissions to a second user to archive process audit data). Before cloning the databases, create the second user in the new schema or database.
   - Use a separate database user (Oracle) or database (SQL Server or DB2) to host the cloned databases.

3 Install 9.9 Products, Prepare Old Environment, Shut Down Old Products, Migrate Database Components, Prepare to Migrate Data, and Upgrade the Software AG Infrastructure

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Gather Values for OneData Installation

- Go to the *old_Software AG_directory* \profiles\ODE\configuration \com.softwareag.platform.config.propsloader directory. Open the com.softwareag.catalina.connector.http.pid-*port* .properties file and the com.softwareag.catalina.connector.https.pid-*port* .properties file and note the port numbers.
- Go to the *old_Software AG_directory* \profiles\ODE\configuration\tomcat\conf directory and open the context.xml file. Note the connection prefix at the beginning of the schema user name (for example, in the metadata schema user name dev2_md, the connection prefix is dev2). Also note the database connection parameters for the metadata, work area, and release area database schemas.

Install the 9.9 Products

Follow the instructions in *Using the Software AG Installer* (October 2015 release) and *Installing Software AG Products* 9.9 to install your 9.9 products in a new installation directory, with the additional guidelines in this section.

If you are upgrading a cluster of Integration Servers or My webMethods Servers, read " Integration Server Clustering Requirements" on page 18 or " My webMethods Server Clustering Requirements" on page 19, respectively, before proceeding.

Run the Software AG Installer

1. For installation directory, specify a new Software AG installation directory.
2. On the product tree, select the 9.9 products to install. Also select Database Configuration.

   Note: In 9.0, it is a Blaze plug-in to Software AG Designer. In 9.5 and later, for compatibility reasons, the Blaze plug-ins are delivered in a self-contained workbench that is based on Eclipse 3.6 and is separate from Software AG Designer 9.5 and later, which are based on Eclipse 4.x.

3. For the product panels, follow the instructions in *Installing Software AG Products* 9.9, with the additional guidelines below.

Note: If you are installing on the same machine as your old release, the installer often allows you to assign ports used by an old product to the new product as well, even if the old product is running. Assigning the same ports means you will not need to edit port values in assets and clients when you begin using the new release.

Product	Guidelines
ActiveTransfer Server	Point the database connection at the ActiveTransfer database component you want to use with ActiveTransfer Server 9.9.
webMethods Broker	Choose to *not* create a Broker Server configuration.
Integration Server	Point the database connection at the ISInternal database component you want to use with Integration Server 9.9.
My webMethods Server	Choose to *not* create a server instance. **Note:** You will migrate old server instances later in this procedure. You will also migrate the MywebMethodsServer database component and then point server instances to that database component later in this procedure.
OneData	Enter the values you noted earlier for ports, repository ID, and repository name, and database component connection. Make sure the database connections point to the OneData database components you want to use with OneData 9.9. **Note:** Reusing port values means that Web service clients of the old OneData will not have to change URLs to access OneData 9.9.
Trading Networks	Point the database connection at the TradingNetworks database component you want to use with Trading Networks 9.9.
Universal Messaging	Choose to *not* create a server instance, Template Applications instance, or Enterprise Manager instance. **Note:** If you install a server, you will migrate old server instances later in this procedure. If you install Template Applications and Enterprise Manager, instances will be created for you later in this procedure.

Complete the 9.9 Installation and Install Latest Fixes on 9.9 Products

Follow the instructions on completing the 9.9 product installations in *Installing Software AG Products* 9.9.

Those instructions include installing the latest fixes on your 9.9 products.

- Install the latest *product* fixes on all 9.9 products except the Infrastructure component named Common Platform.
- For some products, fixes relating to migration are separate from product fixes. Install the latest *migration* fixes on all 9.9 products, including the Common Platform. Migration fix names for the Common Platform follow the convention OGI_9.9_Migration_Fix*number* .
- Install the latest migration framework fix. Fix names for the migration framework follow the convention MIG_9.9_MigrationFramework_Fix*number* and are listed under Common Library.

For instructions on installing fixes, see *Using the Software AG Update Manager* and the fix readme files.

Note: You will install fixes on the Common Platform and on database scripts later.

Note: When you install My webMethods Server, you do not create an instance. The fix is therefore copied to the My webMethods Server installation and gets deployed after you migrate the old server instances later in this guide, when you initialize My webMethods Server.

Perform Final Post-Installation Tasks

If you use My webMethods Server, and you installed the 9.9 release on the same machine as the old release, you removed the environment variable WM_HOME from the global path. Do not redefine the environment variable after installing My webMethods Server 9.9.

If you installed your 9.9 products on a Windows system, and you installed them as Windows services, the default startup mode for the services is Automatic. To prevent the 9.9 products from being started accidentally before they should be started, set the services to Manual instead. If you installed on a UNIX system, and you have scripts that automatically start daemons, disable the scripts for the same reason.

Important: Do not start any 9.9 products at this point, or before migrating database components or data. Do not start 9.9 products before the instructions in this guide explicitly tell you to do so, or your database components could become corrupted.

9.6, 9.7, or 9.8 upgrade: *Installing Software AG Products* 9.9 explains that you can create additional Integration Server instances after installation. However, you can also, or instead, have the Integration Server migration utility create 9.9 instances for you from the old instances (see "Complete the Integration Server and Hosted Wm Packages Upgrade" on page 55).

Prepare the Old Environment for Upgrade

Install Fixes on Old Products

Install the latest product fixes on the old ActiveTransfer, API-Portal, CentraSite, Infrastructure Data Collector, My webMethods Server, Optimize, and Process Engine. For instructions on installing fixes, see *Using the Software AG Update Manager* and the fix readme files. After you install the fix on the old product, start the old product.

Prepare the Old API-Portal

1. Start the old API-Portal Cloud Controller.
2. If the old API-Portal is installed in a clustered, high-availability setup, make sure all nodes are running and accessible to the ZooKeeper ensemble.
3. Register each node with the parent node by running this command:

   ```
   acc> add node logical_node_name IP_address_or_host_name [@port ]
   user_name password
   ```

4. Run this command:

   ```
   acc> startall
   ```

5. Back up the old master tenant to a file.
   - 9.7 upgrade: Run these commands:

     ```
     acc> set acc config backup.restore.tenant.app.types=com.aris.umcadmin.
     y-umcadmin-run-prod
     acc> set acc config backup.restore.tenant.app.type.names=umc
     acc> backup tenant master to full_path_to_backup_file
     username=your_user_name password=your_password
     ```

   - 9.8 upgrade: Run these commands:

     ```
     acc> set acc config backup.restore.tenant.app.types=ECP,ADS,UMC
     acc> backup tenant master to full_path_to_backup_file
     username=your_user_name password=your_password
     ```

6. For each tenant in the old API-Portal, you will have to create a tenant in API-Portal 9.9. List all tenants in the old API-Portal by running this command:

   ```
   acc> list tenants
   ```

7. Back up each old tenant's API and related data; user data; API-Portal document storage data, including all access rights; and collaboration data to a file. The file extension .acb will automatically be added to each file.
   - 9.7 upgrade: Run these commands:

```
acc> set acc config backup.restore.tenant.app.types = com.aris.umcadmin.
y-umcadmin-run-prod, com.aris.modeling.components.y-server-run-prod,
com.aris.adsadmin.y-adsadmin-run-prod, com.aris.runnables.ecp-run-prod
acc> set acc config backup.restore.tenant.app.type.names = umc, abs, ads, ecp
acc> backup tenant tenant_name to full_path_to_backup_file
username=your_user_name password=your_password
```

- 9.8 upgrade: Run these commands:

```
acc> set acc config backup.restore.tenant.app.types=UMC,ABS,ADS,ECP
acc> backup tenant tenant_name to full_path_to_backup_file
username=your_user_name password=your_password
```

8. If you created a customized view of your old API-Portal, back up the view as follows:

   a. Open API-Portal in a browser and log on with Administrator credentials.

   b. Go to the **Administration** page. In the **Views** page under the **Customization** section, hover over the name of the custom view and click **Backup**. API-Portal creates a zip file that contains the customized view.

   c. Save the zip file.

Prepare the Old CentraSite

If you changed settings for CentraSite Control from their defaults, note those settings so you can make the same changes in the new CentraSite installation.

Prepare the Old Software AG Designer

Export Integration Server Definitions

1. In the old Software AG Designer, go to **Window > Preferences**. On the **Preferences** dialog box, in the left navigation bar, go to **Software AG > Integration Servers**.

2. Click **Export** and complete the dialog box. Software AG Designer will save the file with the extension .properties.

Export CloudStreams Server Definitions and Projects

1. To export CloudStreams Server definitions, go to **Window > Preferences**. On the **Preferences** dialog box, in the left navigation bar, go to **Software AG > CloudStreams Servers**. Click **Export** and complete the dialog box. Software AG Designer will save the file with the extension .properties.

2. Open the CloudStreams Development perspective. In the CloudStreams Governance list, right-click **CloudStreamsGovernance projects to export**, click **Export**, and complete the dialog box.

3. Repeat the previous step to export CloudStreams Provider projects.

Export Event Bus Console Configurations

1. In the old Software AG Designer, open the **File > Export** wizard. In the Select panel, go to **Run/Debug > Launch Configurations** and click **Next**.
2. In the tree, select the Event Bus Console configurations. In the **Location** field, specify the directory to which to export. Software AG Designer will save each file with the extension .launch.

Export Preferences

If you want to migrate your preferences to Software AG Designer 9.9, you export them from the old Software AG Designer and then import them into Software AG Designer 9.9. To export the preferences, do the following:

1. In the old Software AG Designer, open the **File > Export** wizard. In the Select panel, go to **General > Preferences** and click **Next**.
2. In the **To preference file** field, specify the name of the file to which you want to export your preferences. Software AG Designer will save the file with the extension .epf.

Prepare the Old Integration Server

1. If you are using webMethods Broker, shut down the new Broker Monitor if it is running. Restart the old Broker Monitor, and then start the old Broker Server.
2. Start the old Integration Server.
3. Suspend document retrieval for all triggers and make sure all guaranteed messages have finished processing, as follows:
   a. Open Integration Server Administrator and point to the old Integration Server.
   b. Go to **Packages > Management**. Click **View Locked Elements**, click **Unlock Elements**, select all elements of triggers, and then click **Unlock Selected Elements**.
   c. 9.0 upgrade: Go to the **Settings > Messaging > Broker/Local Trigger Management** page. In the **Individual Trigger Controls** area, in the **Active** column under **Document Retrieval**, click **edit all** and set **Retrieval State** to **Suspended**. Refresh the page until the **Persisted Queue Counts** field shows 0 for every trigger.
   d. 9.5, 9.6, 9.7, or 9.8 upgrade: Go to the **Settings > Messaging > webMethods Trigger Management** page. In the **Individual Trigger Controls** area, in the **Active** column under **Document Retrieval**, click **edit all** and set **Retrieval State** to **Suspended**. Refresh the page until the **Current Queue Counts** field shows 0 for every trigger.
4. If you are using webMethods Broker, make sure all outbound Broker messages in the client side queue have been sent, as follows:
   a. Make sure Integration Server is connected to the Broker.

b. 9.0 or 9.5 upgrade: Go to the **Settings > Resources > Store Settings** page. In the **Outbound Document Store** area, make sure the **Current Documents in Outbound Store** field shows 0.

c. 9.6, 9.7, or 9.8 upgrade: Go to the **Settings > Messaging > webMethods Messaging Settings** page. Make sure the **CSQ Count** field shows 0 for the Broker connection alias.

> **Note:** The **CSQ Count** field contains the same information as the **Current Documents in Outbound Store** field on the **Settings > Resources > Store Settings** page. You need only check the **CSQ Count** field.

5. 9.6, 9.7, or 9.8 upgrade: If you are using Universal Messaging, make sure all outbound Universal Messaging messages in the client side queue have been sent, as follows:

   a. Make sure Integration Server is connected to each Universal Messaging server that is acting as a webMethods Messaging provider.

   b. Go to the **Settings > Messaging > webMethods Messaging Settings** page. Make sure the **CSQ Count** field shows 0 for the Universal Messaging connection aliases.

6. Make sure all outbound JMS messages in the client side queue have been sent, as follows:

   a. Make sure Integration Server is connected to the JMS providers.

   b. Go to the **Settings > Messaging > JMS Settings** page. In the **JMS Connection Alias Definitions** area, make sure the **CSQ Count** field shows 0 for every JMS connection alias.

7. If you are going to upgrade business processes, and you have business process models that use volatile transition documents and process instances of those models are in a Started state, allow the Started processes instances to complete or suspend them before you upgrade.

> **Important:** If you do not allow Started process instances to complete or suspend before you upgrade, you could lose data.
>
> **Note:** Started process instances of models that use guaranteed transition documents do not have to complete or be suspended before you upgrade.

Prepare the Old My webMethods Server

When you want to edit configuration settings for My webMethods Server, you download the appropriate configuration file from the MywebMethodsServer database component to the My webMethods Server installation and make the changes in that file. You can then either upload the file to the database component and delete it from the file system, or keep it in the file system so its settings will be used in preference to the equivalent settings in the database.

Go to the *old_Software AG_directory* \MWS\server\instance_name \config directory and check for any such files. If you do not want such files to be migrated to the new installation, delete them from the old installation.

Prepare the Old OneData

1. In the old OneData, go to the **Home > Administer > Job Center** page. On the **Filter** page, for each job type, filter by **Active** or **Pending Active**. If a job has one of these statuses, terminate the job or wait for it to complete.
2. If you are using JMS with OneData, go to the *old_Software AG_directory* \profiles \ODE\webapp\onedata\WEB-INF\lib directory and back up all client jar files required for JMS providers.
3. If you are using Kerberos authentication with OneData, back up your Kerberos-based SSO configurations and files.

Go to ***old_Software AG_directory*** **/...**	**Back Up This File**
profiles\ODE\configuration\tomcat\conf	server.xml
profiles\ODE\configuration	custom_wrapper.conf and jaas.config
profiles\ODE\workspace\webapps\onedata \WEB-INF	web.xml
profiles\ODE\configuration \com.softwareag.platform.config.propsloader	All com.softwareag.jaas. realm.pid-*SSO_realm_name*

Prepare the Old Optimize

1. If you want to reuse your Central Configuration System (CCS) Environment definition, export the environment to an XML file. In My webMethods, go to the **Administration > System-Wide > Environments > Define Environments** page. Select the check box next to the environment, click **Export**, and provide a file name and location for the exported environment.
2. If you want to cluster Analytic Engines in the new installation, you will need a Terracotta Server Array. For planning information, see *Getting Started with the webMethods Product Suite and Terracotta* 9.9, *Configuring BAM* 9.9, and the Terracotta BigMemory Max 4.1 documentation.

Shut Down the Old Products

For most products, see the instructions in the table below to shut down on a Windows system. On a UNIX system, use the instructions in the product documentation for your old release.

Running as...	Shut down from...
Services	Windows Services window. Services are listed as Software AG *product release*.
Applications	Windows Start menu. Applications are listed as **Software AG > Stop Servers > *product***.

For the products below, follow these instructions.

Product	Shut Down Instructions
API-Portal	If the old and new API-Portal are on the same machine, run this command in the old API-Portal Cloud Controller: `acc> stopall`
CentraSite	If the old and new CentraSite are on the same machine, shut down all old Application Server Tiers by stopping the Software AG Runtime services, then shut down the old Registry Repository by stopping its service.
EntireX	Shut down EntireX as instructed in the previous table, then shut down all Brokers, RPC Servers, and customer applications that use EntireX libraries. For instructions, see the product documentation for your old release.
Mobile Administrator	Run this command: `/etc/init.d/appfresh-mobile stop`
Terracotta Server Array	Shut down all non-webMethods clients.

Migrate Database Components

The new release might require changes to the database components, such as new tables, columns, keys, or indexes. In this section, you run scripts that update the existing database schemas so they are compatible with the new product release. The scripts might modify the existing database components, or might create parallel database components with the new structure and then insert, select, rename, and drop the tables, columns, keys, and indexes.

Shut Down Products that Connect to Database Components

If you have not already done so, shut down all old ActiveTransfer, Integration Server, My webMethods Server, OneData, Optimize, and Presto instances that connect to database components.

Before You Migrate Trading Networks Database Components

1. Use your data editor to delete all data from the ARCHIVE_WORKTABLE table in the TradingNetworksArchive database component.
2. If you created custom indexes for your Trading Networks database components, check whether those custom indexes conflict with indexes that will be created when you run the Trading Networks migration scripts in the next step. If so, drop the custom indexes. The migration scripts are located in the *9.9_Software AG_directory* \common\db\scripts\RDBMS \tradingnetworks\migrate directory, under the directories listed below.

Old Release	Directories
9.0	\40_to_45 and 45_to_50
9.5 or 9.6	\45_to_50
9.7	\50_to_55
9.8	\55_to_56 and 56_to_65

Special Considerations for Migrating Other Database Components

- If you are using business processes, you have stored information about in-progress, completed, or failed business process instances. This stored information is called business process runtime data. If you do not want to finish running in-progress process instances or resubmit completed or failed process instances from your old installation in your 9.9 installation, do not migrate your business process runtime data (see table below).
- If you are using Optimize with a DB2 RDBMS, there are special considerations for the URL you specify on the database component migration command. You must do the following:

- Specify the schema name in the URL using all capital letters.
- Specify the options `CreateDefaultPackage=true,ReplacePackage=true`, and `DynamicSections=3000`. These settings will affect all database components in the same schema or database.

Install Fixes on Database Scripts

Install the latest fix on the old database scripts using the Software AG Update Manager. The fix names follow the convention DC_*old_release_number* _DBS_Fix*latest_fix_number* . For instructions on installing fixes, see *Using the Software AG Update Manager* and the fix readme file.

Database Component Names

Database component names you might use in the migration command are as follows:

If you are migrating this data...	Specify these *database_component* values
ActiveTransfer Server	`ActiveTransfer`
Business process runtime	`ProcessAudit, ProcessEngine, Archive, Staging, Reporting`
Business rules	`BusinessRules`
Integration Server	`ISInternal, ISCoreAudit, CrossReference, DocumentHistory, DistributedLocking`
My webMethods Server	`MywebMethodsServer`
OneData	`OneDataMetadata, OneDataReleaseArea, OneDataWorkArea`
Optimize	`Analysis, ProcessTracker, ProcessAudit, DataPurge`
Trading Networks	`TradingNetworks, TradingNetworksArchive`

Migrate Database Components

Important: After you migrate database components to 9.9, you can no longer use them with your old environment.

On the machine on which you installed Database Configuration 9.9, open a command window or shell, go to the *9.9_Software AG_directory* \common\db\bin directory, and migrate database components by running the commands below. After you run each command, check the log file dcc_*yyyymmddhhmmss* in the *9.9_Software AG_directory* \common\db\logs directory.

If you are using an Oracle or DB2 RDBMS, and you are not using the default tablespace, also specify the -tsdata *data_tspace_name* and -tsindex *index_tspace_name* parameters.

On an AIX system, enclose the values *“db_server_URL”*, *”existing_db_user”*, and *“password”* in quotation marks.

If you are migrating the Integration Server database components, you must migrate them first.

1. Run this command for each Integration Server database component you need to migrate:

   ```
   dbConfigurator.{bat|sh} -a migrate -d {oracle|sqlserver|db2luw}
   -c database_component_name -v latest -l db_server_URL
   -u existing_db_user -p password
   ```

2. Migrate the rest of the database components.

   - If you want to migrate the rest of your database components at the same time, run this command:

     ```
     dbConfigurator.{bat|sh} -a migrate -d {oracle|sqlserver|db2luw}
     -c all -v latest -l db_server_URL -u existing_db_user -p password
     ```

     Note: Since you already migrated the Integration Server database components, and you most likely do not have all other database components (because you most likely do not have all products), you will see messages in the log file that say migration failed for the database components you do not have. You can ignore these messages.

   - If you want to migrate the rest of your database components individually, run this command for each database component you need to migrate:

     ```
     dbConfigurator.{bat|sh} -a migrate -d {oracle|sqlserver|db2luw}
     -c database_component_name -v latest -l db_server_URL
     -u existing_db_user -p password
     ```

Prepare to Migrate Data

Migration Utilities

Many products offer migration utilities that automatically migrate data from your old installation to your new installation. This section describes the general behavior of the migration utilities; any exceptions to the general behavior are noted in the product-specific chapters.

Some migration utilities for some products run without prompting you for any information. Migration utilities for other products ask you to select the data to migrate, and offer these options:

- You can run a custom migration, in which you select the data to migrate. The utility gathers your settings through a series of prompts, then migrates the selected data. You can export your settings and use them in other upgrades.
- You can run a migration with imported settings. The imported settings can come from settings you exported from a custom migration, or from the default migration provided with the product installation. The settings for default migrations are described in the product-specific chapters.

Note: When you run migration utilities, you provide the full path to the old installation, and sometimes the path to the 9.9 installation. If you supplied a symbolic link as the installation directory when you installed the old or 9.9 product, the path you provide to the migration utility must be the same symbolic link path you supplied during installation.

If a migration utility fails to migrate a piece of data, the utility asks whether to continue with the next piece of data or abort the migration. If you choose to abort, the utility exits. The utility does not revert the 9.9 product installation. You can address the issue and rerun the utility.

Migration utilities write detailed migration information to the command window and to the migrationLog.txt file in the *9.9_Software AG_directory* \install\logs directory. By default, utilities write INFO, ERROR, and FATAL messages to the log. If you want to increase the logging level for a product's migration to DEBUG, go to the product directory that contains the log4j.properties file (for example, the *9.9_Software AG_directory* *product* \bin\migrate or \migrate\bin directory), open the file in a text editor, set the `log4j.logger.Migration` property to `DEBUG, MigrationFile`, and save and close the file.

Create Zip Files as Source of Data to Migrate

If your old and new installations for a product are on different machines, create a Zip file of the old product installation to use as the source of the data to migrate.

Create a Zip File for My webMethods Server

1. Go to the *9.9_Software AG_directory* /MWS/bin/migrate directory and copy the zip-mws.{bat|sh} file to the *old_Software AG_directory* /MWS/bin/migrate or *old_Software AG_directory* /MWS/bin directory.

   Note: The directory you see varies depending on which old release you have. If you installed fixes on the old My webMethods Server, the file might already exist.

2. On the old machine, open a command window or shell, go to the directory that contains the file you just copied, and run the command `zip-mws.{bat|sh}`. The command creates a zip file named mws.zip in *old_Software AG_directory* /MWS/migrate-zip directory.

3. Copy the mws.zip file to any directory on the 9.9 machine that hosts My webMethods Server.

   Important: If using FTP to copy, use the binary file transfer mode/type. If you use another mode/type, the Zip file might become corrupted.

Create Zip Files for All Other Products

The instructions below use the Java Archive tool to create the Zip file. Specify the location of the Java Archive tool in the JAVA_HOME and PATH system variables on the machine that hosts the old product installation. The tool is located in the *Software AG_directory* \jvm\jvm\bin directory.

Note: On some systems, the lower-level jvm directory name includes additional information, such as \jvm\jvm160_32, or \jvm\jvm170, or \jvm\jvm_64.

To create the Zip file for a product, go to the product's old machine and open a command window or shell. Go to the directory indicated in the table below and enter the command indicated in the table. If multiple commands are listed, use the same Zip file name in each command.

Product	Go to this directory...	Run this command...

Command Central	Software AG directory that contains old Command Central	`jar cfM` *`Zip_file_name`*`.zip common\conf`
		`jar ufM` *`Zip_file_name`*`.zip profiles\CCE`
		`jar ufM` *`Zip_file_name`*`.zip install\products`

Product	Go to this directory...	Run this command...
EDA/NERV	Software AG directory that contains old EDA/NERV	jar cfM *Zip_file_name*.zip install \products jar ufM Zip_file_name.zip common\conf jar ufM *Zip_file_name*.zip common \EventTypeStore jar ufM *Zip_file_name*.zip common\nerv \bundles The common\nerv\bundles directory will exist only if you deployed NERV bundles in the old release. If you deployed NERV bundles or the Event Type Store to non-default locations, add the bundles to the Zip file to preserve the file structure shown in the commands above.
Software AG Infrastructure	Software AG directory that contains old Software AG Runtime	jar cfM *Zip_file_name*.zip common \conf install\products profiles \CTP You might see the message profiles\CTP: no such file or directory. You can ignore this message.
Infrastructure Data Collector	Old Infrastructure Data Collector directory	jar cfM *Zip_file_name*.zip common \conf install\products profiles \InfraDC
Integration Server	Old Integration Server directory	jar cfM *Zip_file_name*.zip *
OneData	Software AG directory that contains old OneData	jar cfM *Zip_file_name*.zip profiles/ODE/bin/onedata/config/*
9.6, 9.7, or 9.8 upgrade: Platform Manager	Software AG directory that contains old Platform Manager	jar cfM *Zip_file_name*.zip common \conf jar ufM *Zip_file_name*.zip profiles\SPM jar

Product	Go to this directory...	Run this command...
		`ufM Zip_file_name.zip install \products`
Universal Messaging	Old Universal Messaging directory	`jar cfM Zip_file_name.zip server lib`

Copy the product's Zip file to any directory on the 9.9 machine that hosts the product.

Important: If using FTP to copy, use the binary file transfer mode/type. If you use another mode/type, the Zip file might become corrupted.

Upgrade the Software AG Infrastructure

Infrastructure components are automatically installed with many products. These components are the Common Platform, Software AG Runtime, Security Infrastructure, and Web Services Stack.

Important: The Software AG Infrastructure migration utility supports migration from a Windows system to another Windows system, and from a UNIX system to another UNIX system that is using a JVM from the same vendor. The utility does not support migration from one operating system to a different operating system (for example, does not support Windows system to a UNIX system, or vice versa). For a list of operating systems supported by the 9.9 release, see *System Requirements for Software AG Products*.

Before Migrating Data

Make sure all products in the old and new installations are shut down.

Run the Software AG Infrastructure Migration Utility

The Software AG Infrastructure utility migrates the following:

- The infrastructure configuration.
- If present, the Software AG Runtime configuration.
- Debug, JAAS, JMX, port, security, single-sign on, watchdog, and Web Services Stack configurations; journal log settings; the user repository; and the password store.
- 9.5, 9.6, 9.7, or 9.8 upgrade: Users, groups, and roles.

The Software AG Infrastructure migration utility runs without prompting you for any information. If an error occurs, the utility exits.

For a general description of migration utility behavior, see "Migration Utilities" on page 36.

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \common\migrate\osgi\bin directory, and run this command:

   ```
   migrate.{bat|sh}
   {-srcDir|-srcFile} full_path_to_{old_Software AG_directory|Zip_file}
   -destDir full_path_to_9.9_Software AG_directory
   -importFile 9.9_Software AG_directory\common\migrate\osgi\bin\
   migrate{901|950|960|970|980}sbs.dat
   -silent true
   ```

2. If you see a *Software AG_directory* \profiles\CTP directory in both the old and the 9.9 installation directories, go to the *9.9_Software AG_directory* \profiles\CTP\bin \migrate directory and run the same command again.

3. Install the latest product fix on Common Platform 9.9. The Common Platform fix names follow the convention OGI_9.9_Fix*number* . For instructions, see *Using the Software AG Update Manager* and the fix readme files.

Update the Software AG Runtime NERV Configuration

Go to the *9.9_Software AG_directory* \profiles\CTP\configuration \com.softwareag.platform.config.propsloader directory and open the com.softwareag.platform.eda.nerv.properties file. Make sure the JMS provider references point to your 9.9 JMS provider and the file system paths point to the 9.9 installation. If they do not, update them.

Open the same file in the old installation. If any values in the two files other than the path values mentioned above are different, copy the old value over the 9.9 value.

Where to Go from Here

- If you see product-specific chapters for your products after this chapter, perform the tasks in those chapters in the exact order in which the chapters and tasks are presented in this guide. The order of performance is critical because your products have many inter-dependencies, including a shared infrastructure. Then go to "Complete Final Upgrade Tasks for All Products, and Troubleshooting" on page 115 and complete the tasks in that chapter.
- If you do not see product-specific chapters for your products after this chapter, go to "Complete Final Upgrade Tasks for All Products, and Troubleshooting" on page 115 and complete the tasks in that chapter.

4 Complete the EDA/NERV, Terracotta, and Messaging Upgrades

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Complete the EDA/NERV Upgrade

Run the EDA/NERV Migration Utility

The EDA/NERV migration utility migrates the following:

- Software AG Designer Event Bus Console connection configurations that use NERV.
- NERV configuration bundles, if you deployed any.
- Event Type Store.

The EDA/NERV migration utility runs without prompting you for any information. If an error occurs, the utility exits.

For a general description of migration utility behavior, see "Migration Utilities" on page 36.

On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \common\migrate\EDA\bin directory and run this command:

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Software AG_directory|Zip_file}
-destDir full_path_to_9.9_Software AG_directory
-importFile 9.9_Software AG_directory\common\migrate\EDA
\bin\migrate{901|951|960|970|980}sbs.dat
-silent true
```

Update Paths

Go to the *9.9_Software AG_directory* \common\conf directory and open the com.softwareag.eda.nerv.properties file. Verify that the JMS provider references point to your 9.9 JMS provider and the update file system paths to point to the 9.9 installation. If they do not, update them.

Complete the Terracotta Upgrade

This section explains how to complete an upgrade of a Terracotta Server Array that is deployed in support of webMethods products. For all other setups, see the Terracotta BigMemory Max 4.3 documentation.

Follow the instructions in *Getting Started with the webMethods Product Suite and Terracotta* to install the 4.1.4 or later license key and tc-config.xml file.

You must upgrade all Terracotta Server Array clients and servers in the cluster to 9.9 (clients) and 4.3.1 (servers) before restarting the cluster.

If you have a single server:

1. Shut down your old Terracotta Server Array.
2. Start Terracotta Server Array 4.3.1.

If you have a mirror group consisting of an active server and a mirror server:

1. Shut down the old mirror server.
2. Shut down the old active server.
3. Start active server 4.3.1.
4. Start mirror server 4.3.1.

Complete the Universal Messaging Upgrade

Before Migrating Data

Make sure the old Universal Messaging is shut down.

Run the Universal Messaging Migration Utility

The Universal Messaging migration utility migrates server instances within the old installation to the new installation. This includes migrating each old server instance’s data directory and license file to the corresponding new server instance. The migration utility also creates instances of Template Applications and Enterprise Manager, if you installed them.

For a general description of migration utility behavior, see "Migration Utilities" on page 36. The Universal Messaging migration utility does not offer a default migration.

Perform a Custom Migration

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \UniversalMessaging\tools\migrate directory, and run the command `migrate.{bat|sh}`.
2. The utility asks for the full path to one of the following:
   - The old Universal Messaging installation (for example, C:\old_Software AG_directory \nirvana or UniversalMessaging or /opt/*old_Software AG_directory* / nirvana or UniversalMessaging).

- The Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.

3. The utility asks whether to import migration settings. Enter N.
4. For each server instances in the old Universal Messaging installation, the utility asks the following:
   - The utility asks whether to migrate the server instance.
   - If the utility cannot determine whether the instance was running as an application or service, it asks for that information.
   - If the license is in the default location specified in the Software AG Installer, the utility asks whether to migrate the license. Enter Y.
5. The utility asks whether to export your settings. If you want to perform other migrations by importing the settings from this session, enter Y. If not, enter N.
6. The utility asks whether to begin migration. If you enter Y, the utility migrates the data you selected.

Migrate Using Imported Settings

Imported settings come from settings you exported from a custom migration. These settings are stored in a file named migrate.dat in the *9.9_Software AG_directory* \UniversalMessaging\tools\migrate directory from which you ran the custom migration. Copy the migrate.dat file to any directory on machines that host Universal Messaging 9.9 installations to which you want to migrate data.

Migrating Using Imported Settings with Prompting

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \UniversalMessaging\tools\migrate directory, and run the command `migrate.{bat|sh}`.
2. The utility asks for the full path to the old Universal Messaging installation or to the Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.
3. The utility asks whether to import migration settings. Enter Y and, when prompted, provide the full path to the migrate.dat file.

Migrate Using Imported Settings without Prompting

On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \UniversalMessaging\tools\migrate directory, and run the command below. If an error occurs, the utility exits.

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Universal Messaging_directory|Zip_file}
-importFile full_path_to_migrate.dat -silent true
```

Migrate Custom Changes, Data Directory, and License File

1. In release 9.9, the nserver.conf and nserverdaemon.conf files were combined into a single file named Server_Common.conf. If you made any custom changes to the nserver.conf or nserverdaemon.conf file for an old server instance, make the same changes to the Server_Common.conf file for the corresponding 9.9 server instance. The Server_Common.conf file is stored in the *9.9_Software AG_directory* /UniversalMessaging/server/umserver/bin directory. The nserver.conf and nserverdaemon.conf files are stored in the *old_Software AG_directory* /nirvana/server/umserver/bin directory.
2. If the data directory for an old server instance was not in the default location specified in the Software AG Installer, the migration utility did not migrate the data directory. Copy the data directory from the old server instance to the 9.9 server instance. The default location for the data directory is the *9.9_Software AG_directory* \UniversalMessaging\server\instance_name \data directory. Then edit the file to point to the new server instance's Server_Common.conf file in that directory.
3. If the license file for an old server instance was modified or moved from the default location it was placed in by the Software AG Installer, the migration utility did not migrate the license file. Copy the license file from the old server instance to the 9.9 server instance. The default location for the license file is the *9.9_Software AG_directory* \UniversalMessaging\server\instance_name directory. Then edit the file to point to the new server instance's Server_Common.conf file in that directory.

9.0, 9.5, 9.6 Upgrade: Prepare to Use webMethods Messaging (Native Pub-Sub)

If you plan to use the Integration Server feature called webMethods Messaging (also known as native pub-sub) with Universal Messaging, do the following:

1. Start Enterprise Manager.
2. Select the realm, click the Config tab, and then expand Protobuf Config.
3. Double-click the FilterProtobufEvents property and set it to true.

Complete the webMethods Broker Upgrade

Types of Upgrade

Migrate from webMethods Broker to Universal Messaging

For instructions, see *Migrating from webMethods Broker to webMethods Universal Messaging* 9.9.

Migrate from webMethods Broker to webMethods Broker

Type of Upgrade	Instructions
Installed webMethods Broker 9.6 from the 9.9 release on a machine that is running a different operating system than the old machine.	Create new storage and then migrate Broker Server configuration information from the old storage to the new storage using the webMethods Broker export/import utilities. For instructions, see *Administering webMethods Broker* 9.6.
9.0 or 9.5 upgrade: Installed webMethods Broker 9.6 from the 9.9 release on an IBM System z machine that is running SUSE Linux Enterprise Server or Red Hat Enterprise Linux.	Same as above.
Installed webMethods Broker 9.6 from the 9.9 release on a machine that has the same operating system and hardware as the old machine.	Follow the instructions in "Create Broker Server 9.6 and Use the Old Storage" on page 46.
9.6, 9.7, or 9.8 upgrade: Continue using webMethods Broker 9.6 from the old release.	Point 9.9 Broker clients, including Integration Server 9.9, to webMethods Broker 9.6 in the old installation.

Create Broker Server 9.6 and Use the Old Storage

1. Back up the old data directory.
2. Go to the *old_Software AG_directory* \Broker\bin directory and back up the old awbrokermon.cfg file.

3. Open the awbrokermon.cfg file and note the location of the old data directory as specified on the dataDir parameter.
4. Stop the old Broker Server, go to the *old_Software AG_directory* \Broker\bin directory, and run this command:

   ```
   server_config remove full_path_to_old_data_directory
   ```

   When asked whether to remove the Broker Server from the Broker Monitor configuration, enter Y.
5. If you need to copy the data directory to a new location (for example, because the old and new webMethods Broker installations are on different machines), do the following:

   a. Copy the old data directory to the new location. If any storage files were located outside the data directory (for example, .stor and .data files), also copy those files.

   b. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \Broker\bin directory, and run the command below. If you copied old storage files in the previous step, specify the qs_map_file parameter for each file.

   ```
   server_config relocate full_path_to_new_data_directory
   [-qs_map_file full_path_to_old_file full_path_to_new_file ]
   ```
6. Create a Broker Server 9.6. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \Broker\bin directory, and run the command below. Broker Server 9.6 details are automatically added to the Broker Monitor 9.6 startup configuration. Make sure the host name and port number for Broker Server 9.6 are the same as they were for the old Broker Server.

   ```
   server_config add full_path_to_old_data_directory -k 9.6_license_file
   -p old_release_port
   ```

   Note: If you are migrating on multi-home machines, and the command fails due to a port binding error, see "Migrate on Multi-Home Machines" on page 48.
7. Make sure Broker Server 9.6 is running as follows:

System	Steps
Windows	a. Open the Windows Services window and make sure the status of the **Software AG Broker Server 9.6**(*port*) service is **Started**. b. Right-click the service and click **Properties**. Make sure the **Path to Executable** field points to the awbroker file for 9.6.
UNIX	a. Run the command `ps -ef | grep awbroker`, which prints all running Broker processes including the path to the executable. Make sure the output text shows a line like this: `full_path_to_9.9_Software AG_directory/Broker/bin/awbroker -d full_path_to_old_data_directory`

System	Steps
	b. Make sure the awbroker process is running and that it is running from the 9.6 awbroker file.

Migrate on Multi-Home Machines

If you are migrating from and to multi-home machines (that is, machines that have multiple network interfaces and IP addresses), the Broker `server_config add` command will disregard the migrated Broker Server settings from the old release, ignore specific IP bindings, and try to bind to the IP address 0.0.0.0. If the command fails due to a port binding error, follow the steps below.

1. Go to the *9.9_Software AG_directory* /Broker/bin directory, open the awbrokermon.cfg file, and add the entries below.

   ```
   server-6849.datadir=full_path_to_data_directory
   server-6849.exec=Software AG_directory /Broker/bin/awbroker
   server-6849.version=9.6
   server-6849.port=6849
   ```

2. Go to the *9.9_Software AG_directory* /Broker/data/awbroker*release_number* /default directory, open the awbroker.cfg file, and add the entries below.

   ```
   executable=9.9_Software AG_directory /Broker/bin/awbroker
   version=9.6
   ```

3. Start the Broker Monitor and use the broker_status command to make sure the Broker Server is running and contains history statistics.

5 Complete the My webMethods Server Upgrade

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Before Migrating Data

1. The instructions in this chapter are for upgrading a My webMethods Server installation with one server instance. If you are upgrading a My webMethods Server cluster (that is, multiple My webMethods Server installations, multiple server instances, or both), read " My webMethods Server Clustering Requirements" on page 19 before continuing.
2. Make sure the old My webMethods Server is shut down.

Run the My webMethods Server Migration Utility

The My webMethods Server migration utility migrates server instances within the old installation to the new installation.

For a general description of migration utility behavior, see "Migration Utilities" on page 36. Exceptions:

- The My webMethods Server migration utility does not offer a default migration.
- The log4j.properties file for My webMethods Server is located in the *9.9_Software AG_directory* \MWS\bin\migrate\resources directory.

Perform a Custom Migration

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \MWS\bin\migrate directory, and run the command `migrate.{bat|sh}`.
2. The utility asks for the full path to one of the following:
   a. The Software AG directory that contains the old My webMethods Server installation (for example, C:\old_Software AG_directory or /opt/*old_Software AG_directory*).
   b. The Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.
3. The utility asks whether to import migration settings. Enter N.

4. For each server instance in the old My webMethods Server installation, the utility asks the following:

   a. The utility asks whether to migrate the server instance.

   b. The utility asks for the MywebMethodsServer database component to use with the migrated server instance. If you are using a copy of the database, the utility prompts for the database URL, user name, and password.

      > **Note:** You cannot use a new database that contains no data.

5. The utility asks whether to export your settings from this session. If you want to perform other migrations by importing the settings from this session, enter Y. If not, enter N.

6. The utility asks whether to begin migration. If you enter Y, the utility migrates the data you selected. If migration completes successfully, the utility also does the following:

   - Deletes old configuration files that are not used by My webMethods Server 9.9, and deletes old properties that are not used by My webMethods Server 9.9 from the 9.9 configuration files.

   - If the old My webMethods Server used the embedded database, copies the database tables from the old My webMethods Server to My webMethods Server 9.9 and converts the tables to the new format.

Migrate Using Imported Settings

Imported settings come from settings you exported from a custom migration. These settings are stored in a file named migrate.dat in the *9.9_Software AG_directory* \MWS \bin\migrate directory. Copy the migrate.dat file to any directory on machines that host My webMethods Server 9.9 installations to which you want to migrate data.

Migrate Using Imported Settings with Prompting

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \MWS\bin\migrate directory, and run the command `migrate.{bat|sh}`.

2. The utility asks for the full path to the old Software AG installation directory or to the Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.

3. The utility asks whether to import migration settings. Enter Y and, when prompted, provide the full path to the migrate.dat file.

Migrate Using Imported Settings without Prompting

On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \MWS\bin\migrate directory, and run the command below. If an error occurs, the utility exits.

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Software AG_directory|Zip
file}
-importFile full_path_to_ migrate.dat
-silent true
```

Migrate the My webMethods Server Infrastructure

1. Migrate your JMX port configurations. Open the com.softwareag.jmx.connector.pidport.properties files in the old and 9.9 *Software AG_directory* \profiles\MWS_*instance_name* \configuration \com.softwareag.platform.config.propsloader directories. If any values in the two files are different, copy the old value over the 9.9 value.
2. If you customized the old My webMethods Server Java Service Wrapper, copy the customizations from the old wrapper_conf or custom_wrapper.conf files to the 9.9 custom_wrapper.conf file. The files are located in the *Software AG_directory* \profiles \MWS_*instance_name* \configuration directory.

Initialize a My webMethods Server Server Instance

The initializing process might take several hours.

1. If you customized portlet permissions in the old My webMethods Server installation and want to retain the same permissions after migration, go to the *99_Software AG_directory* \profiles*instance_name* \configuration directory, open the custom_wrapper.conf file, and add this parameter to the Java Additional Parameters section:

   ```
   wrapper.java.additional.number=-DxmlImport.overwrite.permissions=FALSE
   ```

2. Initialize the My webMethods Server 9.9 server instance by going to the *9.9_Software AG_directory* \MWS\bin directory and running this command:

   ```
   mws.{bat|sh} -s instance_name init
   ```

   The 9.9 components are deployed, and then the server instance shuts down automatically. Restart the server instance.

3. If the old and new My webMethods Server installations are on different machines, verify the host name for the new installation as follows:

   a. Log on to a My webMethods Server 9.9 server instance as Administrator and go to the **Administration > My webMethods> Cluster Settings > Advanced Web and Cluster Configuration for MWS** page.

   b. If the host name is not correct in the **Host** and **MWS Front End URL** field, update the host name.

   c. Go to the **Cluster Status and Control** page and restart the server instance.

4. If you added the parameter in step 1 to the custom_wrapper.conf file, remove the parameter from the file.
5. Restart the server instance.

Point My webMethods Server at a Different Database Component

If you first point My webMethods Server at one MywebMethodsServer database component (for example, a cloned database), and then later want to point it at another (for example, a live database), you will need to run the My webMethods Server migration utility again and then initalize My webMethods Server again.

6 Complete the Integration Server and Hosted Wm Packages Upgrade

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Packages

This guide refers to different types of packages for Integration Server, as follows:

- Custom packages. These include packages created by users in Software AG Designer and business process runtime packages generated by users from Software AG Designer.
- Hosted packages. These are packages provided by Software AG on the Software AG Installer, where they are listed under Integration Server on the product selection tree. On the tree, they are listed using their product names. However, in the file system and within Integration Server, they are listed under their package names, and those names start with Wm. For this reason they are also called Wm packages.

  Examples of hosted Wm packages are ActiveTransfer (WmMFT), Mediator (WmMediator), Process Engine (WmPRT), Trading Networks (WmTN), adapters, and eStandards Modules.

  When you installed Integration Server 9.9, the Software AG Installer created a server instance and copied all Wm packages and language packs you selected on the product selection tree into a package repository. The installer let you choose whether to install the selected Wm packages on the server instance. (The installer always installs the language packs you selected on the server instance.)

Before Migrating Data

1. If you are upgrading a Integration Server cluster, read " Integration Server Clustering Requirements" on page 18.
2. If you are upgrading an Integration Server that hosts Deployer, read "Complete the Deployer Upgrade" on page 84 and complete all necessary pre-migration tasks.
3. Make sure the old Integration Server is shut down.
4. The migration utility will scan the old installation for custom packages. However, it will not find custom packages whose names start with Wm, as this naming convention is used for packages provided by Software AG. If you have custom packages whose names start with Wm, and you want to migrate them, go to the *9.9_Software AG_directory* \IntegrationServer\bin\migrate directory, open the

packages.cnf file, and add a <value name></value> tag that identifies those custom packages.

> **Note:** To simplify future upgrades, and as a general best practice, do not use the naming convention Wm *name* for custom packages.

Run the Integration Server Migration Utility

The Integration Server migration utility can migrate the following:

- Custom (user-created) packages.
- Starting in 9.5, custom (user-created) jar files.
- Integration Server configuration files.
- If the old Integration Server hosted Wm packages and those Wm packages also exist on Integration Server 9.9, the utility migrates the configuration files for those packages.

9.6, 9.7, or 9.8 upgrade: If the old Integration Server installation has multiple server instances, you run the migration utility once for each server instance you want to migrate. You can have the utility migrate old server instances to 9.9 server instances you created after installation or to new server instances that the utility creates for you.

For a general description of migration utility behavior, see "Migration Utilities" on page 36.

Perform a Custom Migration

1. On the 9.9 machine, open a command window or shell and go to the *9.9_Software AG_directory* \IntegrationServer\bin\migrate directory.
2. Run the command `migrate.{bat|sh}`.
3. The utility asks for the full path to one of the following:
   - The old Integration Server installation (for example, C:\old_Software AG_directory \IntegrationServer or /opt/*old_Software AG_directory* / IntegrationServer).
   - The Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.
4. 9.6, 9.7, or 9.8 upgrade: If the old Integration Server installation has more than one server instance, the utility lists the instances and asks which instance to migrate.

   The utility then asks for the name of a 9.9 server instance to be the target of the migration. If you want the utility to migrate to the 9.9 server instance that was created during installation, enter the name of that instance. If you want the utility to create a 9.9 instance and migrate to the new instance, enter a name that does not exist in the 9.9 installation.

If Wm packages exist on the old server instance, and the same Wm packages exist in the 9.9 package repository, the utility will install those Wm packages from the 9.9 package repository on the 9.9 server instance. If language packs exist in the package repository, the utility will install those language packs on the 9.9 server instance.

> **Note:** The migration utility can only install the ActiveTransfer Support package and the Application Platform Support package on the server instance that was created during installation.

5. The utility asks whether to import migration settings. Enter N.
6. The utility asks whether to migrate custom packages. If you choose to migrate all custom packages, the utility migrates the following:

   - Custom packages.
   - The WmChemPayloads, WmPapinetPayloads, WmRNPips, WmFINMessages, and WmFIXMessages packages, if present. These packages contains IS documents for the corresponding eStandards Modules and the schemas for those documents.

   If you choose to migrate selected packages only, the utility lists each of the packages above and asks whether to migrate it.
7. The utility asks whether to migrate Integration Server configuration files. If you choose to migrate selected configuration files only, the utility lists each configuration file and asks whether to migrate it. There are about 50 configuration files.

   If you are using a cloned database with Integration Server 9.9, you must respond N to all prompts that list the Integration Server config\jdbc files. If you are using the live database with Integration Server 9.9, respond Y to these prompts.

   If the Integration Server hosts Trading Networks, and you are using a cloned database with Trading Networks 9.9, you must respond N to all prompts that list the Trading Networks JDBC files TN.xml and *pool_alias_name_in_TN.xml* . If you are using the live database with Trading Networks 9.9, respond Y to these prompts.
8. The utility asks you to specify the behavior to use for new properties that have been added to Integration Server 9.9 and existing properties that have new defaults in Integration Server 9.9. If you choose to specify the behavior to use, the utility lists each property and asks whether to use new behavior or preserve existing behavior. You can also read about these properties in the Integration Server readme.
9. If the Integration Server hosts Wm packages, the utility asks whether to migrate the data (that is, configuration files) for those packages.
10. 9.5, 9.6, 9.7, or 9.8 upgrade: The utility asks whether to migrate custom jar files. If you choose to migrate selected custom jar files only, the utility lists each custom jar file and asks whether to migrate it.

11. If the Integration Server hosts Trading Networks, the utility asks whether to migrate Trading Networks information. If you enter Y, the utility asks the following:

    a. Whether to migrate the Trading Networks configuration file. If you enter Y, the utility copies the configuration properties from the old Trading Networks installation directory, adds them to the new configuration properties in Trading Networks 9.9, and replaces the properties.cnf file in the *9.9_Software AG_directory* \IntegrationServer\instances\instance_name \packages\WmTN \config directory.

    > **Note:** If you have a cluster of Trading Networks instances, you must migrate the configuration file for each instance.

    b. Whether to migrate Trading Networks data. This migration maps the data in the Trading Networks database to the new table structure.

    > **Note:** If you have a cluster of Trading Networks instances, the data is shared by all instances, so only migrate the data for one instance.

    c. Whether to migrate Trading Networks dashboard data. If you enter Y, the utility purges records from the dashboard tables and populates records from BIZDOC tables into dashboard tables. The dashboard tables include TransactionSummaryData, CustomAttributeVolumeValue, TransactionSuccessFailedData, SuccessFailedChartDocIdMap, and TransactionLateFAData.

12. The utility asks whether to export your settings. If you want to perform other migrations by importing the settings from this session, enter Y. If not, enter N.

13. The utility asks whether to begin migration. If you enter Y, the utility migrates the data you selected. If migration completes successfully, the utility also performs the actions in "Other Actions Performed by the Migration Utility" on page 60.

Migrate Using Imported Settings

Imported settings can come from the following:

- Settings you exported from a custom migration. These settings are stored in a file named migrate.dat in the *9.9_Software AG_directory* \IntegrationServer\bin\migrate directory from which you ran the custom migration. Copy the migrate.dat file to any directory on machines that host Integration Server 9.9 installations to which you want to migrate data.

- Settings in the default migrations provided with Integration Server. For each old release, the settings are stored in a file named migrate*release* sbs.dat file in the *9.9_Software AG_directory* \IntegrationServer\bin\migrate directory. The settings tell the migration utility to migrate the data listed under "Run the Integration Server Migration Utility" on page 57. For Integration Server properties that are new or that have new defaults, the default migration chooses the behavior that best preserves backwards compatibility.

Migrate Using Imported Settings with Prompting

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \IntegrationServer\bin\migrate directory, and run the command `migrate.{bat|sh}`.
2. The utility asks for the full path to the old Integration Server installation or to the Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.
3. 9.6, 9.7, or 9.8 upgrade: If the old Integration Server installation has more than one server instance, the utility lists the server instances and asks which server instance to migrate from.
4. The utility asks for the name of a 9.9 server instance to be the target of the migration. See this step in "Perform a Custom Migration" on page 57 for detailed information on what to enter and how the utility proceeds.
5. The utility asks whether to import migration settings. Enter Y and, when prompted, provide the full path to the migrate.dat file or the migrate{901|951|960|970|980}sbs.dat file.

Migrate Using Imported Settings without Prompting

On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \IntegrationServer\bin\migrate directory, and run the command below. The `-instanceName` option specifies both the old server instance to migrate from and the 9.9 server instance to migrate to; if the 9.9 server instance does not exist, the utility will create it. If an error occurs, the utility exits.

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Integration
Server_directory|Zip_file}
-instanceName instance_name
-importFile full_path_to_{migrate.dat|migrate{901|951|960|970|980}sbs.dat}
-silent true
```

Other Actions Performed by the Migration Utility

When migration completes successfully, the migration utility also does the following:

- Deletes old configuration files that are not used by Integration Server 9.9, and deletes old properties that are not used by Integration Server 9.9 from the 9.9 configuration files.
- If the old Integration Server used the embedded database, the utility copies the database tables from the old Integration Server to Integration Server 9.9 and converts the tables to the new format.
- Adds a new property named "Validate schemas using Xerces" to existing Web service descriptors, and sets the new property to the value the watt.server.wsdl.validateWSDLSchemaUsingXerces parameter had in the old Integration Server. The new property replaces the functionality provided by

that parameter. For information about the new property, see *webMethods Service Development Help* 9.9.

- If the old Integration Server was part of a cluster that used Coherence as its caching software, the utility removes the clustering settings so you can reconfigure the cluster to use a Terracotta-based solution. Support for Coherence was discontinued in release 9.0. For more information, see *webMethods Integration Server Clustering Guide* 9.9.
- If you are migrating Business Rules data, the utility upgrades Business Rule projects. You might see XML parsing messages due to a Java bug; you can ignore these messages.
- If you are upgrading CloudStreams, the utility migrates configuration artifacts related to administering CloudStreams Server.

Complete the Integration Server Upgrade

Migrate the Integration Server Infrastructure

If you customized the old Integration Server Java Service Wrapper, copy the customizations from the old wrapper_conf or custom_wrapper.conf files to the 9.9 custom_wrapper.conf file. The files are located in the *Software AG_directory* \profiles \IS_*instance_name* \configuration directory.

Customize Integration Server Startup Files

The Integration Server startup scripts were changed in release 9.7. If you customized the old scripts, see *webMethods Integration Server Administrator's Guide* and the Integration Server readme for instructions on how to duplicate those customizations for 9.9.

Update WSDLs

If you have Provider Web services that have an operation with field names starting with `xml` in the input signature, output signature, header, or faults, do the following:

1. Start Integration Server 9.9.
2. Review the migrationLog.txt file in the *9.9_Software AG_directory* \install\logs directory. If you see this error:

   ```
   A property watt.server.xml.ncname.encode.backward.compatibility exists in
   config/server.cnf with value as true. Make sure you make the required changes
   as specified in the upgrade documentation. Not doing so could have adverse
   effects as support for this property may be dropped in a future release.
   ```

   Update your WSDLs as follows:

   a. Open Integration Server Administrator and point to Integration Server 9.9.

b. Go to the **Settings > Extended** page. If you have the extended setting watt.server.xml.ncname.encode.backward.compatibility and it is set to true, reset it to false.

c. Regenerate the clients for all Provider Web services that have an operation with field names starting with `xml` in the input signature, output signature, header, or faults.

Synchronize and Update Serial webMethods Messaging Triggers

If you are using Integration Server 9.9 with Universal Messaging 9.9, each webMethods Messaging trigger with serial processing must have a corresponding priority named object on Universal Messaging. However, each serial trigger you migrated from your old Integration Server corresponds to a shared named object on Universal Messaging. Create the corresponding priority named object by doing one of the following:

- If Universal Messaging 9.9 is a fresh installation (that is, was not migrated from an earlier release), synchronize the publishable document types with the provider using Software AG Designer or the built-in service pub.publish:syncToProvider.
- If your Universal Messaging 9.9 was migrated from an earlier release, this Integration Server is the only Integration Server that connects to Universal Messaging, and the Shared Client Prefix property for the Universal Messaging connection alias is set to No, disable and enable the Universal Messaging connection alias used by the trigger.
- If your Universal Messaging 9.9 was migrated from an earlier release, and this Integration Server is not the only Integration Server that connects to Universal Messaging or the Shared Client Prefix property is set to Yes, wait till the named object is fully drained and no new documents will be sent to it, and then use Universal Messaging Enterprise Manager to delete the named object. You may need to quiesce document publishers before deleting the named object. Then disable and enable the Universal Messaging connection alias used by the trigger.

Connect Integration Server Functions to Database Components

Connect Integration Server and the Wm packages it hosts to their database components. For instructions, see *Installing Software AG Products* 9.9.

Update Host Names

If your old and new Integration Server installations are on different machine, update the database components below to reflect the 9.9 machine.

If you had scheduled tasks that executed on Integration Server on specific machines in your old installation, update the following:

Database Component	Table	Column
ISInternal	IS_USER_TASKS	TARGET

If you want to be able to resubmit services that ran before you upgraded, update the following:

Database Component	Table	Column
ProcessAudit	WMRULEDIST	SERVERID
ISCore Audit, ProcessAudit	WMDOCUMENT, WMSERVICE, WMSERVICE4X, WMSERVICEACTIVITYLOG, WMSERVICEASSOC, WMSERVICECUSTOMFLDS	SERVERID

Update the Integration Server NERV Configuration

1. Go to the *9.9_Software AG_directory* \profiles\IS_*instance_name* \configuration \com.softwareag.platform.config.propsloader directory and open the com.softwareag.platform.eda.nerv.properties file. Update JMS provider references to point to your 9.9 JMS provider. Update file system paths to point to the 9.9 installation.
2. Open the same file in the old installation. If any values in the two files other than the path values mentioned above are different, copy the old value over the 9.9 value.

9.0 Upgrade: Migrate Custom Jar Files

If you copied external jar files for use with Integration Server 9.0 or any products it hosts, and you want to use them in your 9.9 installation, copy them to the *9.9_Software AG_directory* \IntegrationServer\instances\instance_name\lib\jars\custom directory.

Complete the ActiveTransfer Upgrade

For instructions and details on all steps below, see *Managing File Transfers with webMethods ActiveTransfer* 9.9.

1. Copy keystore files and the security.cnf file from the old installation to the ActiveTransfer 9.9 installation. Update the keystore file paths.

2. Make sure that port values, host names and IP addresses, and local and remote file paths in event file operations and the virtual file system (VFS) are correct for the ActiveTransfer 9.9 installation.
3. If you used temporary accounts and shared files with the old ActiveTransfer, you can copy that information so you can use it with ActiveTransfer 9.9. Go to the *old_Software AG_directory* \IntegrationServer\instances\instance_name directory for the Integration Server instance that hosts ActiveTransfer and copy the TempAccounts directory to the same location in the 9.9 installation.

Complete the CloudStreams Upgrade

If the old Integration Server hosted CloudStreams and you installed the same release or a later release of CloudStreams on the current version of Integration Server, you migrated data and configuration settings for CloudStreams as part of the Integration Server migration. However, you must also complete the tasks below.

1. Set the Integration Server property watt.server.http.listRequestVars.
   a. Open Integration Server Administrator and point to Integration Server 9.9.
   b. Go to the **Settings > Extended > Show and Hide Keys** page.
   c. Select the check box for the watt.server.http.listRequestVars property and click **Save Changes**.
   d. Click **Edit Extended Settings** and set watt.server.http.listRequestVars=asNeeded.
2. Back up any custom packages that contain CloudStreams artifacts.
3. Run the public service `pub.cloudstreams.migration:migrate`, which is available in the WmCloudStreams package. The service migrates old CloudStreams artifacts in custom packages that depend on the WmCloudStreams package and updates them to be compatible with CloudStreams 9.9. Artifacts include SOAP and REST connector services and connections. The utility logs the results of migration to the Integration Server server log.

   The input signature for the service is as follows:

Parameter	Description
allPackages	**Boolean string**. Optional. Set to: ■ `true` to migrate CloudStreams artifacts from all custom packages that are dependent on WmCloudStreams. ■ `false` to migrate CloudStreams artifacts from only custom packages specified on the *packages* parameter. This is the default.

Parameter	Description
packages	**Object**. String array containing the names of custom packages to migrate. You must specify at least one package name.

The output signature of the service is a parameter named Result that consists of an array of iData records. The array contains a record for each custom package specified in the input signature. The fields in each record are as follows:

Field	Description
packageName	**String**. Name of the custom package.
success	**String**. Value that indicates whether migration succeeded (`true`) or failed (`false`).
message	**String**. Information about the migration, such as number of CloudStreams artifacts found in the package and number of CloudStreams artifacts that were successfully migrated.
info	**Object**. Optional. If the service issued info messages during the migration, string array of the messages.
errors	**Object**. Optional. If the service issued errors during the migration, string array of the errors.

Complete the Trading Networks Upgrade

If you chose to have the Integration Server migration utility migrate the Trading Networks configuration file, go to the *9.9_Software AG_directory* \IntegrationServer \instances*instance_name* \config\caching directory and delete the SoftwareAG-IS-TN.xml file.

Complete Adapter Upgrades

If the old Integration Server hosted adapters and you installed the same release or a later release of those adapters on Integration Server 9.9, you migrated data for those adapters as part of the Integration Server migration. For the adapters below, however, you must also complete the indicated tasks. Then restart Integration Server.

For instructions on upgrading adapters when you are not also upgrading the host Integration Server, see the adapter product documentation.

webMethods Adapter for JDBC

If the adapter connects to an Oracle RDBMS using an OCI driver, set these environment variables on the Integration Server 9.9 machine:

Platform	Environment Variable Setting
Solaris	LD_LIBRARY_PATH=/ORACLE_HOME/lib
HP	SHLIB_PATH=/ORACLE_HOME/lib
AIX	LIBPATH=/ORACLE_HOME/lib
Linux	LD_LIBRARY_PATH=/ORACLE_HOME/lib

webMethods Package for Microsoft .NET

If you modified the wmSystem.exe.config file in the old installation, make the same changes to the same file in the new installation. The file is located in the *Software AG_directory* \IntegrationServer\DotNetHost directory.

webMethods PeopleSoft EnterpriseOne Adapter

9.0 upgrade: If the adapter connects to an Oracle RDBMS, go to the *old_Software AG_directory* \IntegrationServer\lib\jars directory and copy the Oracle JDBC driver file to the *9.9_Software AG_directory* \IntegrationServer\instances\instance_name \lib\jars \custom directory.

webMethods SAP Adapter

1. 9.0 upgrade: Go to the *old_Software AG_directory* \IntegrationServer\lib\jars directory and copy all files whose file name contains the phrase *sapjco*.* to the *9.9_Software AG_directory* /IntegrationServer/instances/*instance_name* /lib/jars/custom directory.
2. Go to the specified directory below:
   - 9.0 or 9.5 upgrade: *old_Software AG_directory* \IntegrationServer\packages \WmSAP\ns\wm\sap\internal\la\routingListener
   - 9.6, 9.7, or 9.8 upgrade: *old_Software AG_directory* \IntegrationServer\instances *instance_name* \packages\WmSAP\ns\wm\sap\internal\la\routingListener

Copy the node.ndf file to the *9.9_Software AG_directory* \IntegrationServer\instances *instance_name* \packages\WmSAP\ns\wm\sap\internal\la\routingListener directory.

3. If the Integration Server 9.9 machine uses UNIX, set these environment variables on the machine:

Platform	Environment Variable Setting
Linux, Solaris	LD_LIBRARY_PATH
HP	SHLIB_PATH
AIX	LIBPATH

4. If the Integration Server 9.9 machine uses Windows, go to the C:\WINDOWS \system32\drivers\etc directory on the old Integration Server machine, open the services file, and copy all entries that contain the phrase "sapgw" into the same file in the 9.9 installation. Examples of these entries are sapgw00 3300/tcp, sapgw01 3301/ tcp, sapgw02 3302/tcp, sapgw03 3303/tcp, sapgw04 3304/tcp, and sapgw05 3305/tcp.

Complete eStandards Module Upgrades

If the old Integration Server hosted eStandards Modules and you installed the same release or a later release of those modules on Integration Server 9.9, you migrated data for those modules as part of the Integration Server migration. For the modules below, however, you must also complete the indicated tasks. Then restart Integration Server.

For instructions on upgrading eStandards Modules when you are not also upgrading the host Integration Server, see the eStandards Modules when product documentation.

webMethods SWIFTNet Module

Reconfigure the module. For instructions, see *webMethods SWIFTNet Module Installation and User's Guide*.

7 Complete the Software AG Designer, Business Process Runtime, and Blaze Upgrades

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Migrate Software AG Designer Data

Before Migrating Data

1. Open Software AG Designer 9.9 and point to a new 9.9 workspace. For example, you can accept the default workspace99.
2. Install any third-party features you need (for example, support for Subversion). The Eclipse release installed with Software AG Designer 9.9 is Eclipse 4.4.2, so make sure any features you add are compatible with Eclipse 4.4.2.
3. If you are upgrading Events, go to the **File > Import** wizard. In the Select panel, go to **Run/Debug > Launch Configurations** and click **Next**. In the **From Directory** field, go to the directory to which you exported your Event Bus Console configurations in Export Event Bus Console Configurations and select all files with the extension .launch. Click **Finish**.
4. If you exported your preferences before upgrading, import them as follows:
    a. Go to the **File > Import** wizard. In the Select panel, go to **General > Preferences** and click **Next**. In the **From preference file** field, specify the .epf file to which you exported your preferences. Click **Finish**.

    b. Go to **Window > Preferences**. In the preferences window, go to the **Java > Installed JREs > Execution Environments** page, if multiple JREs are listed, make sure JRE 1.8 is selected as the default, or remove older JREs from the list. Also update settings that point to old product installations to point to 9.9 product installations instead. For example, for My webMethods Server, update the **Server > Runtime Environments** settings. For Application Platform, update the path to the runtime instance.

Migrate Application Platform, Business Process, and Business Rule Projects

1. Start Software AG Designer 9.9.
2. If your projects are stored in a source control system, use the import wizard for that system (for example, **Import SVN > Project from SVN**). If your projects are stored in the Software AG Designer workspace, do the following:

   a. Go to the **File > Import** wizard.

   b. In the Select panel, go to **General > Existing Projects into Workspace** and click **Next**.

   c. In the Import Projects panel, do the following:

      a. Click **Select Root Directory** and go to the workspace that contains a type of project you want to import, or click **Select Archive File** and go to the directory that contains a type of project you want to import.

      b. In the **Projects** box, select the projects to import. Select **Copy projects into workspace**. Click **Finish**.

3. Verify the following:

   - All Application Platform projects compile without errors. If errors appear in the **Problems** tab, click each error and resolve it. For example, if you see errors indicating that classes cannot be resolved to a type, edit the project's class path configuration.

   - All imported business process projects appear in the **Solutions** tab under the Processes node.

   - All imported business rules projects appear in the **Solutions** tab under the Rules node.

4. If you imported business rule projects, right-click each project in the Rules Explorer and then click **Upgrade Project**.

Migrate Task Application Projects

1. In Software AG Designer 9.9, go to **Window > Preferences > Server > Runtime Environments**. If the **Installed server runtimes** list does not include a My webMethods Server 9.9, add one.

2. Go to **Software AG > Task Development**. In the preferences window, make sure all your other task-related Software AG Designer preferences are correct, and then click **OK**.

3. Go to the **File > Import** wizard.

   a. In the Select panel, go to **Software AG > Existing CAF Projects into Workspace** and then click **Next**.

   b. In the Import Projects panel, click **Select Root Directory** and go to the workspace or source control system that contains your task application projects, or click **Select Archive File** and go to the directory that contains your task application projects. In the **Projects** box, select the projects to import.

   c. Select **Copy projects into workspace** and then click **Finish**.

4. Make sure all imported task application projects appear in the **Solutions** tab, under the Tasks node.

5. If errors appear in the **Problems** tab, click the **Navigator** tab. Right-click each task application project, click **CAF Tools**, and click **Repair CAF Project**. If errors still appear, restart Software AG Designer.
6. Publish the migrated task application projects to My webMethods Server 9.9. For instructions, see *webMethods BPM Task Development Help* 9.9.

Migrate CloudStreams Server Definitions and Projects

1. Start Software AG Designer 9.9.
2. Import CloudStreams Server definitions as follows:
    a. Go to the **CloudStreams Development** perspective, then go to **Window > Preferences**. On the **Preferences** dialog box, in the left navigation bar, go to **Software AG > CloudStreams Servers**.

    b. Click **Import**, select the .properties file you exported, and click **Open**. Software AG Designer asks whether to overwrite existing servers. Click **OK**, and then click **OK** again to close the **Preferences** dialog box.
3. Import CloudStreams Governance projects as follows:
    a. Go to the **File > Import** wizard. In the Select panel, go to **Software AG > CloudStreams Governance Project** and then click **Next**.

    b. In the Import CloudStreams Governance Project panel, specify the directory that contains the projects you exported in the **Select Root Directory** field. In the **Projects** box, select the projects to import.

    c. Select **Copy projects into workspace** and then click **Finish**.

    d. In the **CloudStreams Development** perspective, make sure all imported governance projects appear in the **CloudStreams Governance** tab.
4. Repeat the previous step to import CloudStreams Provider projects.

Migrate Integration Server Definitions

1. In Software AG Designer 9.9, go to **Window > Preferences**. On the **Preferences** dialog box, in the left navigation bar, go to **Software AG > Integration Servers**.
2. Click **Import**, select the .properties file you exported, and click **Open**. Software AG Designer asks whether to overwrite existing servers. Click **OK**, and then click **OK** again to close the **Preferences** dialog box.

Complete the Business Process Runtime Upgrade

Note: If you have running process instances that are based on a process model created before you upgraded to 9.9, and you now want to regenerate that

> process model in Software AG Designer 9.9, change the version number of the process model to the next sequence number before regenerating it. If you do not do so, the process instances will not behave as expected. When you enable the new version of the process model for execution in My webMethods Server, you will be asked if you want to upgrade running processes; respond No.

1. If you upgraded webMethods Broker, make sure the new Broker that will be used by the Process Engines has the same name in 9.6 that it had in the old release. To do so, open Integration Server Administrator and point to Integration Server 9.9. Go to the **IS Settings > Messaging > webMethods Messaging Settings** page and check the Broker name.

2. If you migrated process runtime packages (see "Run the Integration Server Migration Utility" on page 57), make sure the packages exist in the *9.9_Software AG_directory* \IntegrationServer\instances\instance_name \packages directory. The package names are the project names or custom names you specified in Software AG Designer.

3. Open Integration Server Administrator and point to Integration Server 9.9. Go to the **Settings > JDBC Pools** page and connect the ProcessEngine and ProcessAudit functions to their database components. For instructions, see *Installing Software AG Products* 9.9.

4. If your old and new product installations are on different machines, the Integration Server 9.9 that hosts Monitor 9.9 is not connected to any messaging product and has no process model fragments, and you want to be able to resubmit processes and process steps that ran before you upgraded, follow the steps below.

   a. In Integration Server Administrator, go to the **Package > Management** page and click 🏠 for the WmMonitor package.

   b. Clear the **Resubmit to local IS** check box, click **Submit**, and then reload the package.

   c. In your database editor, go to the PRA_PROCESS_STEP database table in the ProcessAudit database component and update the SERVERID column to point to the Integration Server 9.9 that hosts Monitor 9.9.

5. Make sure document retrieval for all webMethods Messaging triggers is enabled on Integration Server 9.9. For instructions, see *webMethods Integration Server Administrator's Guide* 9.9.

Migrate Closed Loop Analytics

In "Migrate Software AG Designer Data" on page 70 you migrated all assets involved in Closed Loop Analytics. You must now update the JMS topic in Universal Messaging from which Closed Loop Analytics collects escalation events, as follows:

1. Start the Universal Messaging 9.9 server instance you are using for BPM, and then start the Enterprise Manager.

- In the left panel, open the Realms tree and click the server instance you are using for BPM.
- In the right panel, click the JNDI tab and open the Topics tree. Delete the topic for the escalation event type {http://namespaces.softwareag.com/EDA/WebM/Rules/CLA/1.0}ProcessEscalation.

2. In the file system of the machine that hosts the Universal Messaging 9.9 server instance, open a command line window and create a new topic for the escalation event type by entering the command below. Type the command on one line with a space at each of the line breaks shown below.

```
9.9_Software AG_directory/jvm/jvm/bin/java-jar
9.9_Software AG_directory/common/lib/event-routing-jms-util.jar
bind nsp://locahost:9000
{http://namespaces.softwareag.com/EDA/WebM/Rules/CLA/1.0}ProcessEscalation
```

Complete the Blaze Rules Upgrade

Migrate Blaze rules using the instructions in *Using Blaze Rules with BPM and CAF* 9.9.

8 Complete the Optimize Upgrade

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Configure and Start the Terracotta Server Array

If you are clustering Analytic Engines in release 9.9, you must configure and start the Terracotta Server Array. For instructions, see *Getting Started with the webMethods Product Suite and Terracotta* 9.9, *Configuring BAM* 9.9, and the Terracotta BigMemory Max 4.1 or later documentation.

Update Connection to Other Products

Update Connection to Process Engine

If you are using Optimize for Process, do the following:

1. Open Integration Server Administrator and point to an Integration Server 9.9 that hosts a Process Engine.
2. Go to the **Packages > Management** page and click 🏠 for the WmPRT package.
3. Click **Settings** in the left navigation bar and then click **Edit Process Engine Settings**.
4. In the **JMS Server URL** field, identify the JMS provider host name and port as follows:

JMS Provider	URL Format
webMethods Broker	broker://*host* :*port* /*Broker_name* . The default port is 6849.
Universal Messaging	nsp://*host* :*port* . The default port is 9000.

5. Click **Submit** and then reload the WmPRT package.
6. Repeat these steps for every Integration Server that hosts a Process Engine.

Update Connection to Optimize Support Package

If you are using Optimize built-in services, do the following:

1. Open Integration Server Administrator and point to the Integration Server 9.9 that hosts an Optimize Support package.
2. Go to the **Packages > Management** page and click 🏠 for the WmOptimize package.
3. Identify the Analytic Engine host machine and port. The default port is 12503.
4. In the **JMS Server URL** field, identify the JMS provider host name and port as follows:

JMS Provider	URL Format
webMethods Broker	broker://*host* :*port* /*Broker_name* . The default port is 6849.
Universal Messaging	nsp://*host* :*port* . The default port is 9000.

5. Click **Submit** and then reload the WmOptimize package.
6. Repeat these steps for every Integration Server that hosts an Optimize Support package.

Complete the Infrastructure Data Collector Upgrade

9.7 Upgrade: Start Monitoring 4.1 or Later Terracotta Server Arrays

For instructions on each step, see *Administering webMethods Optimize* 9.9.

1. Go to **Applications > Administration > Infrastructure Components > Discovery**.
2. Delete all Terracotta (Asset) entries.
3. Rediscover 4.1 or later Terracotta Server Arrays you want to monitor.
4. Go to **Applications > Administration > Infrastructure Components > Monitored Components**. Select Terracotta Server Array monitors, components, and KPIs.

Run the Infrastructure Data Collector Migration Utility

The Infrastructure Data Collector utility migrates asset configuration files and SNMP, JAAS, JMX, SSH, security, Java Service Wrapper custom wrapper, log, platform debug, Web Services Stack, and watchdog configurations for Infrastructure Data Collector.

For a general description of migration utility behavior, see "Migration Utilities" on page 36.

Before Migrating Data

1. Make sure the old Infrastructure Data Collector is shut down.

2. If you want to migrate SNMP asset configuration files, go to the *9.9_Software_AG_directory* infrastructuredc\bin\migrate\old_release directory and open the snmpMigration.properties file. Type the full paths to the old and new SNMP configuration directories. The contents of the file show the format to use.

Perform a Custom Migration

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \Infrastructuredc\bin\migrate directory and run the command `migrate.{bat|sh}`.
2. The utility asks for the full path to one of the following:
   - The Software AG directory that contains the old Infrastructure Data Collector installation (for example, C:\old_Software AG_directory or /opt/*old_Software AG_directory*).
   - The Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.
3. The utility asks whether to import migration settings. Enter N.
4. The utility asks whether to migrate SNMP asset configuration files. If you respond Y, the migration utility will migrate the old files you identified in the snmpMigration.properties file.
5. The utility asks whether to export your settings. If you want to perform other migrations by importing the settings from this session, enter Y. If not, enter N.
6. The utility asks whether to begin migration. If you enter Y, the utility migrates the data you selected.

   If you chose to migrate asset configuration files, and a file that has the same name as an old file already exists in the Infrastructure Data Collector 9.9 installation, the utility backs up the 9.9 file to the *9.9_Software AG_directory* \infrastructuredc \migrationbackup directory before migrating the old file. The same is true for all migrated directories and files.

Migrate Using Imported Settings

Imported settings can come from the following:

- Settings you exported from a custom migration. These settings are stored in a file named migrate.dat in the *9.9_Software AG_directory* \infrastructuredc\bin\migrate directory from which you ran the custom migration. Copy the migrate.dat file to any directory on machines that host Infrastructure Data Collector 9.9 installations to which you want to migrate data.
- Settings in the default migrations provided with Infrastructure Data Collector. For each old release, the settings are stored in a file named migrate*release* sbs.dat file in the *9.9_Software AG_directory* \infrastructuredc\bin\migrate directory. The settings

tell the migration utility to migrate the data listed in "Complete the Infrastructure Data Collector Upgrade" on page 77.

Migrate Using Imported Settings with Prompting

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \infrastructuredc\bin\migrate directory, and run the command `migrate.{bat|sh}`.
2. The utility asks for the full path to the old Infrastructure Data Collector installation or to the Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.
3. The utility asks whether to import migration settings. Enter Y and, when prompted, provide the full path to the migrate.dat file or the migrate{901|951|960|970|980}sbs.dat file.

Migrate Using Imported Settings without Prompting

On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \infrastructuredc\bin\migrate directory, and run the command below. If an error occurs, the utility exits.

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Infrastructure Data
Collector_directory|Zip_file}
-importFile full_path_to_{migrate.dat|{901|951|960|970|980}sbs.dat}
-silent true
```

Deploy the Optimize Environment

Unless otherwise noted, see *Configuring BAM* 9.9 for detailed instructions on the Optimize steps in the sections below.

Not Reuse the Old Optimize CCS Environment Definition

If you do not want to reuse your old Central Configuration System (CCS) Environment definition, start My webMethods Server 9.9, open My webMethods, and configure your Optimize 9.9 environment.

For Infrastructure Data Collector, when you define hosts, enter the same value in the **Host Name or IP Address** field of the Add/Edit Host dialog box that you had in the old release.

> **Note:** If the 9.9 value is not identical to the old value, you will have to stop monitoring existing assets, rediscover them, and reselect them for monitoring after you finish upgrading.

Skip to the step for deploying the Optimize 9.9 environment, below.

Reuse the Old Optimize CCS Environment Definition

To reuse the old CCS environment definition that you exported to file before installing Optimize 9.9, follow the steps below.

1. Start My webMethods Server 9.9 and open My webMethods.
2. Import the old CCS environment, as follows:
   1. Go to the **Applications > Administration > System-Wide > Environments > Define Environments** page.
   2. Click **Import Environment**. On the Import Environments dialog box, click **Browse**, go to the file to which you exported the old CCS environment, and click **OK**. When prompted, confirm the migration.
   3. When asked "Do you wish to migrate the environment to the latest version?", click **Yes**. The CCS imports the old environment and migrates it to Optimize 9.9 Logical Servers.
3. Update the database pool connection settings for the database pool definition associated with the environment you imported, as follows:
   1. Go to the **Applications > Administration > System-Wide > Environments > Database Pool Configuration** page.
   2. In the **Pool** column, click the imported database pool.
   3. In the **Pool Settings** area, set the **Minimum Connections** field to 8, and the **Maximum Connections** field to 60.
   4. Make sure the database connection information is correct. In the **Database Connection** area, click **Test** and make sure the message `Test Passed` displays at the bottom of the page.
   5. Click **Save**.
4. Go to the **Applications > Administration > System-Wide > Environments > Define Environments** page. Under **Environment Name**, click the environment you imported.
5. If you are using a different JMS provider with the 9.9 release than you used with the old release, do the following:
   1. Click the **Configure Servers** tab. Under **Configuration**, open the **Default Settings** tree and click **JNDI Configuration**.
   2. In the **Naming Factory Type** list, click the JMS provider you are using with the 9.9 release. If you are using webMethods Broker, complete the **Broker Name** field. If you are using Universal Messaging, clear the field.
   3. Click the **Map Endpoints** tab. In the first **Incoming Connections** field for the JMS Provider, type `broker` if you are using webMethods Broker or `nsp` if you are using Universal Messaging. In the second field, type the port number.

   d. Click **Save**.

6. Validate the environment as follows:

   a. Click each tab that shows the icon ○ except the **Validate** tab. If you change any settings on a tab, and the tab has a **Save** button, click **Save**.

   b. When all tabs except the **Validate** tab show ✓, click the **Validate** tab.

7. Click **Finish**.

Deploy the Optimize 9.9 Environment

1. Start the Optimize 9.9 components you installed. These components can include the Analytic Engine, Infrastructure Data Collector, and the Web Services Data Collector.
2. Start the JMS provider you are using with the 9.9 release.
3. Deploy your Optimize 9.9 environment. For instructions, see *Configuring BAM* 9.9.
4. In My webMethods, go to the **My webMethods > System Settings > Servers** page.
   - If you have BPMS installed, select the **BPM and BAM** option and identify the upgraded Analytic Engine and the Integration Server that hosts Monitor. You can identify the hosts using DNS name or IP address.
   - If you have BPMS installed, but are not using BPM (that is, you are using only Optimize), select the **BAM only** option and identify the upgraded Analytic Engine. You can identify the host using DNS name or IP address.

   Click **Check Server Status** and make sure the server is available (green icon). It might take some time for the server to become available (for example, 15 minutes). Then click **Save**.

Switch from webMethods Broker to Universal Messaging

If you were using webMethods Broker as your JMS provider in the old release, and are using Universal Messaging in 9.9, do the following:

1. Start Universal Messaging and open Universal Messaging Enterprise Manager.
2. Connect to a server instance, then click it.
3. Click the **Config** tab. Expand **Global Values** and set AllowRealmAdminFullAccess to true.

9 Complete the Asset Build Environment, Deployer, and Application Platform Upgrades

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Complete the Asset Build Environment Upgrade

Go to the *old_Software AG_directory* \common\AssetBuildEnvironment\master_build directory and open the build.properties file. Copy the property values to the same file in the 9.9 installation.

Complete the Deployer Upgrade

You have two options for migrating Deployer data:

- You can migrate all Deployer global (default) settings, server aliases, target groups, and projects using the Integration Server migration utility.
- You can choose whether to migrate global (default) settings, and you can choose which server aliases, target groups, and projects to migrate using the Deployer migration feature.

The instructions for each option are provided below.

For instructions on tasks you must perform in Deployer, see *webMethods Deployer User's Guide*.

Important: Migrating Deployer projects does not convert the assets in those projects from their old release to 9.9. You must migrate all product assets as instructed earlier in this guide before migrating Deployer settings, server aliases, target groups, and projects.

Before Migrating Deployer Settings, Server Aliases, Target Groups, and Projects

1. Make sure you have upgraded all source and target product servers that were defined in the old Deployer to their 9.9 releases, and have migrated all product data and assets to the 9.9 releases, as instructed in this guide.
2. Make sure you have upgraded the Integration Server 9.9 that hosts Deployer 9.9 and migrated its data using the instructions in this guide.

3. Make sure you are a member of the Administrators ACL, or of the Internal, Developer ACLs, and DeployerAdmin ACLs, for both the Integration Server that hosts the old Deployer and the Integration Server that hosts Deployer 9.9. For instructions, see *webMethods Integration Server Administrator's Guide* 9.9.

4. In the old Deployer, edit all connections to old source and target product servers to point to the new 9.9 source and target product servers. You do not have to supply user names and passwords if you do not know them.

5. If you installed Deployer 9.9 on the same machine as the old Deployer, you might want Deployer 9.9 to use the ports that were used by the old Deployer. If so, do the following:

   a. In Integration Server Administrator, point to the Integration Server that hosts the old Deployer and go to the **Security > Ports** page.

   b. Click **Add Port**, click **webMethods/HTTP**, and click **Submit**. Enter the port details, click **Yes** for **Enable**, click **WmRoot** under the package name, and click **Save Changes**.

   c. Click **Change Primary Port**, click the new port, and click **Update**.

   d. In the **Access Mode** column of the new port, click **Edit**. In the **Edit Access Mode** page for the port, click **Set Access Mode to Allow by Default**. Click **OK** on the warning pop-up. The message `Successfully changed access mode for Port Service Access Settings` appears.

   e. Disable the old port by clicking **Yes** in the **Enabled** column for that port.

Migrate All Deployer Settings, Server Aliases, Target Groups, and Projects Using the Integration Server Migration Utility

In the previous section, you upgraded the Integration Server that hosts Deployer 9.9 and migrated its data. Do not start the host Integration Server before you migrate Deployer settings, server aliases, target groups, and projects. If you do, a file named MIGRATION_DONE is created in the WmDeployer package, and you will not be able to migrate Deployer settings, server aliases, target groups, and projects. If you did start the host Integration Server, delete this file.

Migrate Deployer settings, server aliases, target groups, and projects using the instructions in "Complete the Integration Server and Hosted Wm Packages Upgrade" on page 55.

Migrate Deployer Settings, Server Aliases, Target Groups, and Projects Using the Deployer Migration Feature

Start the Old Deployer and Deployer 9.9 and Migrate Global (Default) Settings

1. Start the old Deployer and Deployer 9.9.

2. In Deployer 9.9, go to the **Tools > Migrate Data** page. In the **Host** and **Port** fields, identify the old Deployer to migrate from. In the **User** and **Password** fields, provide the user name and password to use to log onto the old Deployer.
3. Deployer 9.9 lists the server aliases, target groups, or projects to migrate in the right-hand pane. In the **Page Size** list, click the number of server aliases or projects to list at a time.
4. In Deployer, you set defaults for all projects, such as defaults for dependency checking, project locking, and general deployment. You also set defaults for Integration Server and Trading Networks projects, such as defaults for suspending assets during deployment, overwriting existing assets, and activating assets after deployment. If you want to migrate these default settings, in Deployer 9.9, click **Migrate Default Settings**.

Migrate Server Aliases (Connection Information)

You can choose the server aliases to migrate. However, if you want to migrate a target group, you must either migrate all server aliases in that target group, or remove the server aliases you do not want to migrate from the target group.

1. In Deployer 9.9, click **Migrate Server Aliases**. In the **Select Server** list, click the type of server whose aliases to migrate.
2. In the **Select** column, select the check boxes next to the server aliases you want to migrate. In the **Point Selected Aliases to Servers** list, click **9.9**.
3. Click **Migrate Server Aliases**. Deployer migrates the selected server aliases, then disables those rows and displays **Yes** in the **Migrate** column for the server aliases.
4. Repeat the steps above for each type of server whose aliases to migrate.
5. You cannot migrate Integration Server HTTPS server aliases from Deployer. Manually recreate these server aliases in the host Integration Server.

Migrate Target Groups

1. If you did not migrate all server aliases in the target groups you want to migrate, in the old Deployer, remove those server aliases from the target groups.

   > **Important:** If target groups include invalid server aliases, deployment of projects that use those target groups fails.

2. In Deployer 9.9, click **Migrate Target Groups**. In the **Select Server** list, click the type of server whose target groups to migrate.
3. In the **Simulate Migration** column, click ⚙ in the row for each target group to migrate.

   Deployer displays a message that indicates whether the server aliases that are referenced by the target group have been migrated. If any server aliases have not been migrated, migrate them or remove them from the indicated target group.

4. In the **Select** column, select the check boxes next to target groups to migrate. In the **Point Selected Aliases to Groups** list, click **9.9**.
5. Click **Migrate Target Groups**. Deployer migrates the selected target groups, then disables those rows and displays **Yes** in the **Migrate** column for the groups.

Migrate Deployment Projects

1. Start all 9.9 source product servers referenced by Deployer projects.

> **Note:** If a server referenced by a Deployer project is not running, or the connection settings for a server referenced by a project are not correct, you will have to rebuild the project later. Deployer will display messages to this effect.

2. In Deployer 9.9, click **Migrate Projects**.
3. In the right-hand pane, in the **Simulate Migration** column, click ⚙ row for each project to migrate. Deployer displays a message that indicates whether the source servers that are referenced by the project are running. If the message indicates that any of the referenced source servers are not running, start those servers.
4. In the **Select** column, select the check boxes next to the projects to migrate, and then click **Migrate Projects**. Deployer displays a new window with a dynamic migration report. You can display the migration report for each project in the **Report** column in the original window. You can display the latest migration report by clicking **View Latest Migration Report**.

Complete the Application Platform Upgrade

Rebuild and redeploy your Application Platform projects using Asset Build Environment and Deployer. For instructions, see *webMethods Application Platform User's Guide*.

If your Application Platform projects include configuration properties, go to the *old_Software AG_directory* /profiles/*instance_name* /configuration/ com.softwareag.platform.config.propsloader directory and copy the Application Platform project configuration property files to the same directory in the new installation.

10 Complete the Command Central Upgrade

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Before Migrating Data

Make sure the old Command Central is shut down.

Run the Command Central Migration Utility

The Command Central migration utility can migrate the following:

- Environment configurations, which include metadata about the environment, nodes (installations), and products that Command Central is managing.
- Users, roles, and groups.
- Port configurations.
- 9.5, 9.6, 9.7, or 9.8 upgrade: License reports by license key and templates.
- 9.6, 9.7, or 9.8 upgrade: JAAS configuration and password manager data.
- 9.7 or 9.8 upgrade: ehCache DiskStore.

For a general description of migration utility behavior, see "Migration Utilities" on page 36.

Perform a Custom Migration

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \CommandCentral\migrate\bin directory, and run the command `migrate.{bat|}`.
2. The utility asks for the full path to one of the following:
   - The Software AG directory that contains the old Command Central installation (for example, C:\old_Software AG_directory or /opt/*old_Software AG_directory*).
   - The Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.
3. The utility asks whether to import migration settings. Enter N.

4. The utility asks whether to migrate environment configuration data and users, roles, and groups.
5. 9.5, 9.6, 9.7, or 9.8 upgrade: The utility asks whether to migrate license reports by license key and templates.
6. 9.6, 9.7, or 9.8 upgrade: The utility asks whether to migrate the JAAS configuration and password manager data.
7. 9.7 or 9.8 upgrade: The utility asks whether to migrate ehCache DiskStore data.
8. The utility asks whether to migrate HTTP and HTTPS port configurations.
9. The utility asks whether to export your settings. If you want to perform other migrations by importing the settings from this session, enter Y. If not, enter N.
10. The utility asks whether to begin migration. If you enter Y, the utility migrates the data you selected.

Migrate Using Imported Settings

Imported settings can come from the following:

- Settings you exported from a custom migration. These settings are stored in a file named migrate.dat. The file is stored in the *9.9_Software AG_directory* \CommandCentral\migrate\bin directory from which you ran the custom migration. Copy the migrate.dat file to any directory on machines that host Command Central 9.9 installations to which you want to migrate data.
- Settings in the default migrations provided with Command Central. For each old release, the settings are stored in a file named migrate*release* sbs.dat file in the *9.9_Software AG_directory* \CommandCentral\migrate\bin directory. The settings tell the migration utility to migrate the data listed under "Run the Command Central Migration Utility" on page 90.

Migrate Using Imported Settings with Prompting

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \CommandCentral\migrate\bin directory, and run the command `migrate.{bat|}`.
2. The utility asks for the full path to the Software AG directory that contains the old Command Central installation or to the Zip file you made in "Create Zip Files as Source of Data to Migrate" on page 36.
3. The utility asks whether to import migration settings. Enter Y and, when prompted, provide the full path to the migrate.dat file or the migrate{901|951|960|970|980}sbs.dat file.

Migrate Using Imported Settings without Prompting

On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \CommandCentral\migrate\bin directory, and run the command below. If an error occurs, the utility exits.

```
migrate.{bat|}
{-srcDir|-srcFile} full_path_to_{old_Software AG_directory|Zip_file}
-importFile full_path_to_{migrate.dat|migrate{901|951|960|970|980}sbs.dat}
-silent true
```

Migrate the Command Central Infrastructure

1. If you customized the Command Central Java Service Wrappers, copy the customizations from the old wrapper_conf or custom_wrapper.conf files to the 9.9 custom_wrapper.conf file. The files are located in the *Software AG_directory* \profiles \CCE\configuration directory.
2. For each case below, open the file in the indicated directory in the old installation and in the 9.9 installation. If a value in the old file is different from the corresponding value in the 9.9 file, copy the old value over the 9.9 value. In addition, update any absolute file system paths in the 9.9 files to point to the 9.9 installation. Do not update paths that use tokens (for example, do not update @path\:sag.install.area/common/conf/platform_truststore.jks).

Configuration	File Name in Old and 9.9 Software AG Directories
Debug	\profiles\CCE\configuration \com.softwareag.platform.config.propsloader \com.softwareag.platform.debug.properties
JMX ports	\com.softwareag.jmx.connector.pid-*port* .properties
Security authorization	\com.softwareag.security.authz.pid.properties
Security LDAP	\com.softwareag.security.ldap.pid.properties
Single sign-on service	\com.softwareag.sso.pid.properties
SSH	\com.softwareag.sshd.pid.properties
Watchdog	\com.softwareag.config.watchdog.pid.properties

Configuration	File Name in Old and 9.9 Software AG Directories
Journal logger	\profiles\CCE\configuration\logging\log_config_.xml
9.0 or 9.5 upgrade: Password manager	\profiles\CCE\configuration\security\passman \scPassMan.config.xml

3. 9.0 or 9.5 upgrade: For each type of configuration below, copy the file from the indicated directory in the old installation to the same directory in the 9.9 installation.

Configuration	File Name in Old and 9.9 Software AG Directories
JAAS	\profiles\CCE\configuration\jaas.config file
Password manager	\profiles\CCE\configuration\security\passman\ defaultPassStore.dat and mpw.dat

4. For each case below, open the file in the indicated directory and update file system paths to point to the 9.9 installation.

Configuration	File Name in 9.9 Software AG Directories
JAAS	\profiles\CCE\configuration\jaas.config

Complete the Command Central Upgrade

9.6, 9.7, or 9.8 Upgrade: Restore Trusted Authentication for JAAS

If you chose to migrate your JAAS configuration, follow the steps below to restore trusted authentication.

1. Go to the *9.9_Software AG_directory* \profiles\CCE\configuration\security directory and open the /jaas.config file.
2. Make sure the entries below exist in the Default login context.

```
com.softwareag.security.idp.saml.lm.JMXDelegatedAuthLoginModule sufficient;
com.softwareag.security.idp.saml.lm.SAMLAssertIssuerLoginModule optional;
```

Update the Command CentralNERV Configuration

Go to the *9.9_Software AG_directory* \profiles\CCE\configuration \com.softwareag.platform.config.propsloader directory and open the com.softwareag.platform.eda.nerv.properties file. Update JMS provider references to point to your 9.9 JMS provider, if they do not already. Update file system paths to point to the 9.9 installation.

Configure Platform Manager

Configure Platform Manager on each 9.9 machine from Command Central. For instructions, see *Software AG Command Central Help*.

11 Complete the API-Portal and CentraSite Upgrade

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Complete the API-Portal Upgrade

1. Make sure the old API-Portal is shut down.
2. If you installed API-Portal 9.9 in a clustered, high-availability setup, make sure all nodes are running and accessible to the Zookeeper ensemble. Then register all the nodes with the parent node by invoking the REST service shown below. The default for the load balancer port is 18003.

Parameter	Value
End point	http://*load_balancer_host* :*load_balancer_port* /acc/rest/nodes/
Method type	POST
Content type	application/json
Payload	{"nodename": "*host_name* ", "hostname": "*host_name* ", "port": "*cloud_agent_port* ", "username": "Cloud", "password": "g3h31m“}

3. Start the API-Portal 9.9 Cloud Controller.
4. Run this command:

```
acc> startall
```

5. Restore the old master tenant into API-Portal 9.9 from the backup file you created earlier.

    - 9.7 upgrade: Run these commands:

    ```
    acc> set acc config backup.restore.tenant.app.types = UMC
    acc> restore tenant master from full_path_to_backup_file
    username=your_user_name password=your_password
    ```

    - 9.8 upgrade: Run these commands:

    ```
    acc> set acc config backup.restore.tenant.app.types=ECP,ADS,UMC
    acc> restore tenant master from full_path_to_backup_file
    username=your_user_name password=your_password
    ```

6. Earlier in this guide, you generated a list of tenants in the old API-Portal. For each old tenant on the list except the default tenant and the master tenant, create a corresponding tenant in API-Portal 9.9 by running this command:

   ```
   cmd> cd 9.9_Software_AG_directory \API_Portal\server\bin\work\
   work_apiportalbundle_m\tools\bin
   cmd> y-tenantmgmt.bat  -s http://load_balancer_host :load_balancer_port
   -t tenant_name createTenant -u your_user_name -p your_password
   ```

7. In the User Management Component at http://*host* :*port* /umc, import the license for each tenant you created.

8. In the API-Portal 9.9 Cloud Controller, run these commands:

   ```
   acc> stop apiportalbundle_m
   acc> start apiportalbundle_m
   ```

9. For each new tenant you created, restore the data for the corresponding old tenant into API-Portal 9.9 from the backup file you created earlier by running the commands below. Each file has the file extension .acb.

   Important: Inform API-Portal users that they cannot work on the tenants during the restoration process.

   Important: If the new tenant contains any data, the data will be replaced by the data you restore from the old tenant.

   ```
   acc> set acc config backup.restore.tenant.app.types = UMC, ABS, ADS, ECP
   acc>  restore tenant tenant_name from full_path_to_backup_file
   username=your_user_name password=your_password
   ```

10. Run these commands:

    ```
    acc> stop apiportalbundle_m
    acc> start apiportalbundle_m
    ```

11. If you created a customized view of your old API-Portal, you backed up the view. For instructions on restoring the customized view from the backup, see the backup and restore section in the *webMethods API-Portal Customization Guide*.

Complete the CentraSite Upgrade

Before Migrating Data

1. Make sure the old CentraSite and CentraSite 9.9 are shut down.

2. Export configuration data from the old CentraSite and assets from the old Registry Repository into a Zip file. On the old machine, open a command window or shell, go to the *old_Software AG_directory* /CentraSite/utilities directory, and run this command:

   ```
   sbsExport.{cmd|sh} /full_path_to_Zip_file
   ```

   An example of this command is `./sbsExport.sh /tmp/sbs_cs82_data.zip`

3. If the old and new CentraSite installations are on different machines, copy the Zip file to any directory on the 9.9 machine.

Migrate the Data

CentraSite scripts migrate configuration data from the old CentraSite and assets from the old Registry Repository to the new CentraSite installation. They also transform and migrate the LDAP configuration from the old Registry Repository to the new CentraSite JAAS configuration.

1. Import configuration data and assets from the Zip file you created earlier. Open a command window or shell, go to the *9.9_Software AG_directory* /CentraSite/utilities directory, and run this command:

   ```
   sbsImport.{cmd|sh} /full_path_to_Zip_file
   ```

   An example of this command is `./sbsImport.sh /tmp/sbs_cs82_data.zip`

2. Start CentraSite 9.9.

3. 9.0, 9.5, or 9.6 upgrade: Transform and migrate the LDAP configuration to the new JAAS configuration. From the *9.9_Software AG_directory* /CentraSite/utilities directory, run this command:

   ```
   CentraSiteCommand.{cmd|sh} generate JaasConfiguration -url CentraSite_9.9_URL
   -user CentraSite_9.9_admin_user -password password
   ```

   The script generates JAAS LoginModule entries that correspond to the old LDAP configuration and saves the entries in the jaas.config file in the *9.9_Software AG_directory* /CentraSite/profiles/CTP/configuration directory. For each LDAP domain, the script creates user and group files that map internal (CentraSite) properties to external (LDAP) properties and saves the files in the *9.9_Software AG_directory* /CentraSite/profiles/CTP/configuration/com.softwareag.platform.config.propsloader directory.

4. If you use single sign-on with CentraSite, do the following:

   a. Open the jaas.config file in the old and new *Software AG_directory* /CentraSite/profiles/CTP/configuration directories. Copy the following from the old file to the 9.9 file:

      - ServletHeaderLoginModule for extracting the user ID from the incoming HTTP header.
      - SimpleNameMappingLoginModule, if you are using it.
      - Any other entries you are using to process the extracted user ID.

      The 9.9 jaas.config file should look like this:

      ```
      CentraSite {
      com.softwareag.centrasite.security.cache.ShortTermTokenLoginModule sufficient;
       com.softwareag.security.jaas.login.internal.InternalLoginModule sufficient
       ...
       com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule sufficient
       ...
       com.softwareag.security.jaas.login.modules.ServletHeaderLoginModule
      ```

```
required
 ...
 com.softwareag.security.jaas.login.modules.SimpleNameMappingLoginModule
required
 ...
 com.softwareag.security.sin.is.ldap.lm.LDAPLoginModule required
 ...
};
```

The ShortTermTokenLoginModule establishes delegated authentication in CentraSiteSite to perform secured internal communication. The initial InternalLoginModule is normally only for users in the INTERNAL domain, and the initial LDAPLoginModule is for LDAP users that are logging in directly and not via single sign-on. If you need only single-sign on logins, you can remove the initial InternalLoginModule and LDAPLoginModule.

b. Set up your LDAP configuration to resolve the extracted user ID via LDAP. Modify the generated LDAP login module to enable single sign-on-related options, such as technical user. Apply LDAP single sign-on technical user credentials if necessary.

Complete the Upgrade

1. Change any settings in CentraSite Control that you noted in Prepare the Old CentraSite .
2. If you installed plug-ins that are GUI extensions for CentraSite Control in the old CentraSite installation, install them in the CentraSite 9.9 installation.
3. If you are using API-Portal with CentraSite, republish all API-Portal instances you created in the old release to API-Portal 9.9. For instructions, see the CentraSite documentation.

12 Complete the Mobile Upgrade

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

9.7 or 9.8 Upgrade: Complete the Mobile Designer Upgrade

If you created apps using the Mobile Development perspective in Software AG Designer and want to continue to do so, or if you created apps using Mobile Designer but want to use the Mobile Development perspective in Software AG Designer from now on, delete the environment variable MOBILE_DESIGNER as instructed below.

If you created apps using Mobile Designer and want to continue to do so, and if you also want to continue to use your existing build environment (that is, an IDE other than Software AG Designer, or Jenkins), delete the MOBILE_DESIGNER environment variable as instructed below. Then pass the property `#Denv.MOBILE_DESIGNER=`*`full_path_to_Mobile Designer`* to Ant when calling Mobile Designer Ant targets (for example, enter `ant #Denv.MOBILE_DESIGNER=C:\SoftwareAG\MobileDesigner +Multi-Build`).

To remove the MOBILE_DESIGNER environment variable, do the following:

System	Steps
Windows	Go to the Control Panel, click **System**, and then click **Advanced System Settings**. On the Advanced tab, click **Environment Variables** and delete the MOBILE_DESIGNER environment variable. Then restart your system.
Mac OS X	Delete the MOBILE_DESIGNER environment variable from your .bash_profile, your /etc/launchd.conf file, or both. The file or files that contain the environment variable depend on the version of MAC OS X you are using, and on modifications you might have made. Then restart your system.

Complete the Mobile Administrator Upgrade

Overview

You will complete the Mobile Administrator upgrade by:

- Running the Mobile Administrator updater to upgrade the Mobile Administrator source code and migrate the Mobile Administrator database.
- Verifying that all features required by your use case are configured.
- Updating any build nodes you used with the old Mobile Administrator.
- Updating the App Store Client.

Run the Updater and Verify Configurations

1. Go to the *Software AG_directory* /MobileAdministrator directory and run the updater by running the command `./update.sh` as root.
2. From the same directory, start Mobile Administrator by running the command `/etc/init.d/appfresh-mobile start`.
3. In Mobile Administrator, click **Settings**, click **Sites**, and click a domain. On the domain page, click **Configuration Assistant**. Make sure all the features you need are configured. If necessary, follow the instructions to configure any features that are missing. Repeat this step for each domain.

Update Build Nodes

1. If you used the Mac OS build node with the old Mobile Administrator, update the build node as follows:
   a. Go to the **Mobile Administrator > Build Nodes** page at https://*Mobile Administrator_URL* /build_nodes. Download the appfresh-build-node-*version* .zip file, and extract the files to any directory.
   b. Launch the build node application. On the **Install** tab, in the **Build Node** field, make sure the displayed build node version corresponds to the version of the build node you just downloaded (that is, appfresh-build-node-*version* .zip).
2. If you used the Windows 8 build node with the old Mobile Administrator, update the build node as follows:
   a. Go to the Windows Control Panel and uninstall the previous build node.
   b. Go to the **Mobile Administrator > Build Nodes** page at https://*Mobile Administrator_URL* /build_nodes. Download and run the AppFreshBuildNode-*version* .exe file.
   c. Launch the build node application. On the **Build Node** tab, start the build node by clicking **Start**.

Update the App Store Client

1. Go to the https://*Mobile Administrator_URL* /apps/appfresh-mobile-client/build page.

2. On the **Build Configurations** tab, start the build jobs for the App Store Client by selecting the platforms for which to update the client and clicking **Launch Selected Build Configs**.
3. On the **Build Jobs** tab, verify that the build jobs have completed successfully. Corresponding build nodes should appear on the https://*Mobile Administrator_URL* / build_nodes page. If they do not, click on the build job to display the details that will help you diagnose the problem.

13 Complete the OneData Upgrade

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Before Migrating Data

Make sure the old OneData is shut down.

Run the OneData Migration Utility

The OneData migration utility migrates configuration data.

1. On the 9.9 machine, open a command window or shell, go to the *9.9_Software AG_directory* \profiles\ODE\bin\migrate directory, and run the command below. If an error occurs, the utility exits.

   ```
   migrate.{bat|sh}
   {-srcDir|-srcFile} full_path_to_{old_Software
   AG_directory|Zip_file}
   -silent true
   ```

   Before migrating data, the utility backs up the files that contain the 9.9 configuration data in the *9.9_Software AG_directory* \profiles\ODE\bin\onedata\config directory. The backup files have the extension .bck.

Complete the OneData Upgrade

1. If you customized the old OneData Java Service Wrapper, copy the customizations from the old wrapper_conf or custom_wrapper.conf files to the 9.9 custom_wrapper.conf file. The files are located in the *Software AG_directory* \profiles \ODE\configuration directory.
2. If you are using JMS with OneData, copy the client jar files you backed up earlier to the *9.9_Software AG_directory* \profiles\ODE\webapp\onedata\WEB-INF\lib directory.
3. If you are using Kerberos authentication with OneData, reconfigure it using the backups of the web.xml, server.xml, jaas.config, custom_wrapper.conf and com.softwareag.jaas.realm.pid-*SSO_realm_name* .properties files you created earlier. For instructions, see *Administering webMethods OneData*.

14 Complete the Presto Upgrade

Read and Complete All Previous Chapters

Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.

Overview

This chapter explains how to complete an upgrade from Presto 3.6, 3.7, 3.8, or 3.9 to Presto 9.9. If you want to upgrade from releases of Presto prior to 3.6, contact Software AGGlobal Consulting Services.

Presto stores all of its data, including mashups, mashables, apps and workspaces, in the Presto repository. To upgrade, you will set up Presto 9.9 to use the old Presto repository. If you deployed extensions to features of the old Presto (for example, if you used scripting in mashups and the scripts are stored in files), you will also copy the extensions to the Presto 9.9 installation.

Set Up the Presto Repository

Set Up Repository When Using the Embedded Derby Database

If you used the embedded Derby database for your old Presto repository, follow the instructions below.

1. Presto 9.9 comes with a prepopulated Derby Presto repository in the prestorepository directory. Go to the *9.9_Software AG_directory* /Presto/apache-tomee-jaxrs/bin and rename the prestorepository directory (for example, rename the directory prestorepository_orig).
2. 3.7, 3.8 or 3.9 upgrade: Go to the *old_Software AG_directory* /Presto/apache-tomee-jaxrs/bin directory and copy the prestorepository and mashzonerepository directories to the *9.9_Software AG_directory* /Presto/apache-tomee-jaxrs/bin directory.
3. 3.6 upgrade: Go to the Presto3.6.1/Presto-3.6.1-20130930/prestorepository/derby directory and copy the prestorepository directory to the *9.9_Software AG_directory* / Presto/apache-tomee-jaxrs/bin directory.

Set Up Repository When Using Any Other Database

If you used MySQL, Oracle, PostGres, or SQL Server for your old Presto repository, follow the instructions below.

1. Go to the *old_Software AG_directory* /Presto/apache-tomee-jaxrs/lib directory and copy the JDBC driver jar file for your database to the same directory in the 9.9 installation.

Database	JDBC Driver Jar File
MySQL	mysql-connector-java-5.1.31.jar
Oracle	ojdbc6.jar
PostGres	postgresql-9.2.1004.jdbc4.jar
SQL Server	jtds-1.3.1.jar

2. If you used MySQL, Oracle, or SQL Server for your old Presto, do the following:
   a. Go to the *9.9_Software AG_directory* /Presto/apache-tomee-jaxrs/webapps/presto/WEB-INF/lib directory and delete the jackbe-presto-rds-postgre-derby-9.9.jar file.
   b. Go to the *9.9_Software AG_directory* /Presto/prestorepository directory and copy the jackbe-presto-rds-oracle-mysql-mssql-9.9.jar file to the *9.9_Software AG_directory* /Presto/apache-tomee-jaxrs/webapps/presto/WEB-INF/lib directory.
3. 3.7, 3.8, or 3.9 upgrade: Go to the *old_Software AG_directory* /Presto/apache-tomee/jaxrs/conf directory and copy the tomee.xml file to the same directory in the 9.9 installation.
4. 3.6 upgrade: Configure Presto 9.9 to use the existing repository. For instructions, go to the Presto 9.9 Presto Administration documentation, and then to Getting Started with Presto, and then to Move the Presto and Mashzone Repositories to a Robust Database Solution. Skip steps 1 through 4, and start with step 5.

Update the Presto Configuration and Copy Extensions

Most of the Presto configuration is stored in the Presto repository. This section explains how to copy the parts of the old configuration that were not maintained in the repository, including extensions such as custom security profiles that you deployed in the old Presto. The procedure varies depending on whether you used an external configuration directory with the old Presto.

Update When an External Configuration Directory Was Used

1. Update the classpath in the application server that hosts Presto 9.9 to work with the external configuration directory and any subdirectories. For instructions, see the topic on setting up an external configuration directory in the Presto Administration documentation.

2. Manually migrate the changes you made in the old configuration files listed below from the old Presto to the corresponding files in Presto 9.9. Because of the complexity of the configuration, the easiest method is to use the difference tool offered by many text editors.

   - All files listed in the topic on configuration files that must be internal in the Presto Administration documentation.
   - The userRepositoryLdap.properties file, located in the external configuration directory.

Update When No External Configuration Directory Was Used

1. Copy any extensions you deployed from the old Presto to the corresponding directory for Presto 9.9. See the Presto Administration documentation for a complete list of extensions you might need to copy.
2. Manually migrate the changes you made in the old configuration files listed below from the old Presto to the corresponding files in Presto 9.9. Because of the complexity of the configuration, the easiest method is to use the difference tool offered by many text editors.

   - All files listed in the section on configuration files that must be internal in the Presto Administration documentation.
   - The userRepositoryLdap.properties file, located in the *old_Software AG_directory* / Presto/webapps/presto/WEB-INF/classes directory.

3. If you used database mashables, go to the *old_Software AG_directory* /Presto/apache-tomee-jaxrs/presto/WEB-INF/config/db/jdbc directory. Copy the database driver jar files to the same directory in the 9.9 installation.

Copy Integrated MashZone Data and Execute Upgrade Commands

3.7 or 3.8 Upgrade: Copy Integrated MashZone Data and Execute Upgrade Commands

1. Presto’s Integrated MashZone Server stores additional data. Copy the data to the Presto 9.9 installation as follows:

   a. Go to the *old_Software AG_directory* /Presto/mashzone/data/resources directory and copy the files to the same directory in the 9.9 installation.

   b. Go to the *old_Software AG_directory* /Presto/mashzone/data/jdbcdrivers directory and copy the files to the same directory in the 9.9 installation.

   c. Go to the Admin Console in Presto Hub. Expand the MashZone section, click **Server Settings**, and click the **Resources** tab. Review the paths to resource directories. If any paths point to directories in the old Presto installation, copy those directories to the Presto 9.9 installation and then update the paths in MashZone Administration to point to the correct location in the Presto 9.9 installation.

2. Start the Presto server. Open a command window or shell, go to the *9.8_Software AG_directory* \Presto\prestocli\bin directory, and run these commands:

```
.\padmin.bat runRequestFile -u Administrator -c -w manage
-f ..\..\upgrade\3.9\upgrade_3.9.0.jump.txt
.\padmin.bat importPolicy -u Administrator -w manage
-f ..\..\upgrade\3.9\Presto_RO_Policy.xml
```

3.9 Upgrade: Copy Integrated MashZone Data and Execute Upgrade Commands

1. If you used the Integrated MashZone Feed Editor in Presto 3.9, and you uploaded files or JDBC drivers for use in the Feed Editor, you must copy the files and drivers to the Presto 9.9 installation. In addition, the Feed Editor is now hidden by default and needs to be re-enabled.

   a. Go to the *9.9_Software AG_directory* /Presto/apache-tomee-jaxrs/webapps/presto/WEB-INF/classes directory and open the presto.config file in a text editor. Find the lines shown below.

```
mashzone.administration.disabled = true
mashzone.feededitor.disabled = true
```

   Set the lines to false and save the file.

   b. Go to the *old_Software AG_directory* /Presto/mashzone/data/resources directory and copy the files to the same directory in the 9.9 installation.

   c. Go to the *old_Software AG_directory* /Presto/mashzone/data/jdbcdrivers directory and copy the files to the same directory in the 9.9 installation.

   d. Start the Presto 9.9 server and go to the Admin Console in Presto Hub. Expand the MashZone section, click **Server Settings**, and then click the **Resources** tab. If any paths point to resource directories in the old Presto installation, copy those directories to the Presto 9.9 installation and then update the paths to point to the copied directories.

2. Open a browser and enter the URL `http://`*`Presto_server`*`:`*`Presto_port`*`/presto/hub/dev/`.

3. Log on as administrator. The default user name and password are Administrator/manage.

4. Under the **Pick an API** frame in the left navigation area, expand **PolicyService** then select **reloadPermissionSets**.

5. Click **Run** in the upper right corner. Presto executes the **reloadPermissionSets** API call, which populates the **Request** frame.
6. Copy RAQL user-defined functions deployed in Presto 3.9 to Presto 9.9. Go to the *old_Software AG_directory* /Presto/raql-udfs directory and recursively copy all files other than the SampleRaqlLib file to the same directory in the 9.9 installation.

3.6 or 3.7 Upgrade: Update Mashups that Use RAQL Queries

Presto 3.8 and later contain significant changes and additions to the RAQL query syntax that you use in mashups. If you have mashups from Presto 3.7 or earlier, you must update the mashups that use RAQL queries to run on Presto 9.9, but you might want to also update the other mashups to take advantage of new RAQL query syntax. For complete information, see the topic about migrating old RAQL queries to 9.9 in the Presto documentation.

Update the Snapshots Repository Connection Information

If you used Snapshots in your old Presto and you stored the data in a separate Snapshots repository, follow the steps below.

1. Open the Presto 9.9 Admin Console.
2. If the JDBC driver you used with the old Snapshots repository is not the same driver you used for the old Presto repository, go to the *old_Software AG_directory* /Presto/apache-tomee-jaxrs/lib directory. Copy the JDBC driver jar file for the database that contains the Snapshots repository over the same file in the same directory in the 9.9 installation.

Database	JDBC Driver Jar File
Derby	derby-embedded-10.5.3.jar
MySQL	mysql-connector-java-5.1.31.jar
Oracle	ojdbc6.jar
PostGres	postgresql-9.2.1004.jdbc4.jar
SQL Server	jtds-1.3.1.jar

3. Update the Snapshots 9.9 configuration to point to the Snapshots 9.9 repository as follows:

a. Open the **JDBC Configuration** menu, click **Datasources**, click **SnapshotDatasource**, and then click **Edit**.

b. Edit the configuration properties as needed. For information on the properties, see the section on adding a data source in the Presto Administration documentation.

c. Click **Save**.

15 Complete Final Upgrade Tasks for All Products, and Troubleshooting

Complete Final Upgrade Tasks for All Products

- Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.
- Configure your new products. For instructions, see the 9.9 product documentation.
- If you installed your 9.9 products on a different machine than your old products, make sure to update host names in your 9.9 products, in the connections between your products, and in your database tables. This guide indicates many locations in which to update host names, but make sure you also specify the correct host names when you configure your 9.9 products using the instructions in the product documentation. Also make sure any absolute paths in the 9.9 configuration files point to valid locations, or change them to be correct for the new machine. If the machine has a different operating system or hardware, make sure your JVM settings are correct.
- Read the 9.9 product readmes, including the 9.9 readme for the Software AG Infrastructure, available on the Software AG Documentation website. The product readmes contain this information:
  - Critical information, known issues, fixes, resolved issues that had not been rolled into fixes at the time of the release, and usage notes for your products.
  - Changes to product behavior, services, parameters, properties, APIs, documentation, and terminology. This information is especially important because you might need to modify product files or assets to accommodate the changes.

  The 9.9 readmes contain information for release 9.5 and later; read the information for your old release + 1 through the 9.9 release. For example, if you are upgrading from release 9.0, read the information for releases 9.5 through 9.9.
- Read the product release notes; the release notes provide information on new functionality. Read the information for every release for your old release + 1 through the 9.9 release.
- After installation, you might have set Windows services for products to Manual, and disabled scripts that start UNIX daemons, to avoid automatically starting both old and new products. When your new environment is ready, after you stop running the old products and when you want to start running the new ones, you can reset the Windows services to Automatic, and re-enable the UNIX scripts.

Troubleshooting

If you encounter errors during the upgrade, try the troubleshooting methods below.

- Look in log files.

Type of Log	Location
Installation and uninstallation	*9.9_Software AG_directory* \install\logs directory
Product	In a logs directory in the product file structure, or in the *9.9_Software AG_directory* \profiles*product* \logs directory.
Data migration	*9.9_Software AG_directory* \install\logs directory and product directories indicated by the upgrade procedures
Database migration	*9.9_Software AG_directory* \common\db\logs directory.

- Go to the Knowledge Center on the Empower Product Support Website and search using keywords such as upgrade.
- Look up error messages in the *webMethods Error Message Reference*.
- Go to the Software AG Developer Community and join the upgrade discussion forum.

If you cannot resolve the problem using the methods above, contact Software AG Global Support.