Upgrading Software AG Products On Premises

Version 10.7

October 2020

WEBMETHODS

This document applies to webMethods Migration Framework 10.7 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: SAG-UPGR-107-20231020

Table of Contents

About this Guide

This guide provides upgrade instructions for Software AG products.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	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 and Support

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at http://documentation.softwareag.com.

Software AG Empower Product Support Website

If you do not yet have an account for Empower, send an email to empower@softwareag.com with your name, company, and company email address and request an account.

Once you have an account, you can open Support Incidents online via the eService section of Empower at https://empower.softwareag.com/.

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.

If you have any questions, you can find a local or toll-free number for your country in our Global Support Contact Directory at https://empower.softwareag.com/public_directory.aspx and give us a call.

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.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Upgrades Covered in This Guide

Upgrades Covered in This Guide

This guide explains how to upgrade the following from 9.9 or later to 10.7:

- webMethods products. For BigMemory Max and Terracotta, this guide covers upgrade only when they are being used with other webMethods products.
- Zementis Predictive Analytics when being used with webMethods products.
- Apama and MashZone NextGen.

You can also use the procedures in this guide to migrate data from one 10.7 installation to another 10.7 installation (for example, for a data center move).

Integration Agent does not exist in 10.7. If your old installation included Integration Agent, install Microservices Runtime in your new installation, and then migrate from the old Integration Agent to the new Microservices Runtime using the instructions in this guide for migrating from an old Microservices Runtime to a new Microservices Runtime.

Upgrades and Migrations Covered in Other Guides

The table below lists upgrades that are not covered in this guide, and supplies the correct information source.

Upgrade	Information
From a release prior to 9.9	See *Supported Upgrade Paths*.
webMethods adapters	See the adapter product documentation.
Apama releases earlier than 9.9	See the Apama release notes for the pre-9.9 release.
ApplinX	See the ApplinX documentation.
webMethods Broker to Universal Messaging	webMethods Broker has been deprecated. webMethods Broker 10.7 is a re-release of webMethods Broker 10.5. If you want to migrate to Universal Messaging, first upgrade your other products as instructed in this guide, and then migrate webMethods Broker to Universal Messaging as instructed in *Migrating from webMethods Broker to Software AG Universal Messaging*. If you want to stay on webMethods Broker, first upgrade webMethods Broker as instructed in *Installing and Upgrading webMethods Broker*, and then upgrade your other products as instructed in this guide.
Command Central	See *Software AG Command Central Help*.
webMethods eStandards Modules	See the eStandards Module product documentation.

Upgrade	Information
Mediator to API Gateway	See the *webMethods API Gateway Configuration Guide*.
Presto and MashZone NextGen	There is no direct path from MashZone NextGen 9.9. Instead, you must upgrade to MashZone NextGen 9.10 using the instructions in the 9.10 upgrade guide, and then upgrade from 9.10 using the instructions in this guide.
BigMemory Max when not used with webMethods products	See the BigMemory Max documentation.
BigMemory Max to Terracotta	Terracotta is significantly different from BigMemory Max because Terracotta is designed for storage and caching of typed data, while BigMemory Max is designed for caching of opaque data. Therefore, you cannot directly migrate data and configuration for a BigMemory Max server to a Terracotta server. If you want to migrate, you will have to create custom tooling that takes your particular usage into account.
Zementis Predictive Analytics when not used with webMethods products, and releases prior to 10.1	See the Zementis installation guide for the old release from which you want to upgrade.

Support for Upgrade Across Operating Systems

Integration Server, Microservices Runtime, OneData, and Universal Messaging support migration across machines that have the same operating system and across machines that have different operating systems.

The other products support migration from a Windows system to another Windows system, and from a UNIX system to another UNIX system when both systems are using a JVM from the same vendor. These products do not support migration from one operating system to a different operating system (for example, from a Windows system to a UNIX system, or vice versa).

Release Numbering Exceptions

The products below did not follow general release numbering.

- In the webMethods 9.9, 9.10, and 9.12 releases, ActiveTransfer remained at its 9.8 release.
- webMethods release numbers and BigMemory Max release numbers map as follows: 9.9 (4.3.1), 9.10 (4.3.2), 9.12 (4.3.3), 10.1 (4.3.4), 10.3 (4.3.6), 10.5 (4.3.8), 10.7 (4.3.9).

2 Critical Factors and Requirements for Successful Upgrade

Preparing, Planning, and Testing

The most critical and time-consuming parts of the upgrade process are preparation, planning, and testing. For instructions and guidelines on these topics, see the *Software AG Upgrade Preparation and Planning Guide* and the *Upgrade Testing Guidelines* in *Installation and Upgrade Information for Software AG Products*, a webhelp that contains all installation and upgrade-related documents.

Software AG Tools to Use

Software AG Installer, Software AG Update Manager, and Software AG Migration Utilities

If you want to upgrade using Software AG Installer to install new products; Software AG Update Manager to install fixes; and Software AG migration utilities to migrate product configurations, data, and assets, follow the instructions in this guide.

Command Central

If you want to upgrade products in a single environment using Command Central, follow the instructions in this guide. If you are not using software stacks, you can watch a demo relating to upgrade in the Command Central area of the Software AG TECHcommunity website. You can also read an article on generating keystores and certificates for Command Central in the same area. You can read articles and watch webinars about upgrading with Command Central on the Software AG TECHcommunity website.

If you want to fully automate product upgrade for multiple similar environments using Command Central, see the composite template instructions in *Software AG Command Central Help*. You can reconfigure endpoints such as host names and ports by adding actions in the templates. You will still need to perform the manual tasks in this guide if you cannot script them.

You can use Command Central software stacks to simplify upgrade now and in the future, as the upgrade procedure for stacks is simpler than the procedure for standalone product installations. A software stack is a set of product runtimes and related database components that serve one or more purposes, such as application integration, API management, or business process management. Software stacks enable you to use Command Central to create, monitor, and maintain multiple product installations using bulk operations. The stack upgrade feature is available in the Command Central GUI and lets you automatically generate composite templates that you can then use to upgrade a single environment or to automate the upgrade of multiple environments. Follow the instructions for upgrading software stacks in this guide.

To see which products support upgrade/migration using Command Central, see *Software AG Command Central Feature Support Matrix*.

Command Central supports the upgrade scenarios below.

Same Installation Directory, Hosts, and Ports; Live Database

Command Central renames the old product installation directory, then uses the old directory name for the new product installation.

Benefits: Simple; similar to overinstall in that all paths, hosts, and ports remain the same. Can fully automate using custom composite templates.

Cautions: Risky; rollback is complicated. Significant downtime. Must back up live databases; use of cloned databases is not supported. Requires double disk space because old installation is still present. Requires extensive automated testing before production environments can be migrated.

Command Central provides templates you can use for this scenario on GitHub at https://github.com/SoftwareAG/sagdevops-templates/tree/master/templates/

New Installation Directory, Same Hosts and Ports, Cloned Database

Benefits: New installation is almost identical to old. Safer than first scenario; rollback is much simpler. Can start old installation after shutting down new installation, and vice versa. Can fully automate using custom composite templates or can perform step by step using a combination of the Command Central web user interface and commands.

Cautions: Requires double disk space because old installation is still present. Must adjust third-party tooling to point to new installation directory.

New Installation Directory and Ports, Same Hosts, Cloned Database

Benefits: Safer than first scenario; rollback is much simpler. Downtime is less than first and second scenarios; can migrate gradually while old installation continues to operate. Can fully automate using custom composite templates.

Cautions: Preparation for migration is more complex than first and second scenarios. Requires double disk space because old installation is still present. Must change endpoints to reflect new ports. Must adjust third-party tooling to new installation.

New Hosts, Live or Cloned Database

Typically required because of upgrades to hardware and operating systems. Use of same installation directory and ports can minimize change. Use of DNS to control IP-to-host mapping for the old and new environment can greatly simplify impact on external clients.

Benefits: Safest scenario, and minimal downtime. Easiest scenario when environment runs in a private or public cloud that allows temporary increase of capacity to host new environment and then release resources used by old environment. Can phase migration. Must adjust third-party tooling to new installation. Can fully automate using custom composite templates or can perform step by step using a combination of the Command Central web user interface and commands.

Cautions: Some products allow cross-operating system migration, but for other products, this type of migration might have unpredictable results, especially when old machine is Windows and new machine is non-Windows, or vice versa.

Documentation Needed to Perform the Upgrade

To perform this upgrade, you will need *Installation and Upgrade Information for Software AG Products* for the new release, which combines all installation and upgrade-related documents into a single, easy to use webhelp.

As you progress through the upgrade, you will also need the fix readmes for fixes you install on the new products; these readmes list resolved issues and changes in behavior. You can obtain these from the Software AG Update Manager, or the Empower Fix Explorer.

You might need the documentation listed below to perform this upgrade, depending on the products you are upgrading. Use the documentation for the new release of each product.

- *webMethods API Portal Administrator's Guide* and *webMethods API Portal Customization Guide*
- *webMethods Application Platform User's Guide*
- *webMethods BPM Task Development Help*
- CentraSite documentation
- *webMethods Deployer User's Guide*
- *webMethods Integration Server Administrator's Guide*, *webMethods Service Development Help*, and *webMethods Integration Server Clustering Guide*
- *MashZone NextGen Administration Guide* and *MashZone NextGen User and Developer Guide*
- *Administering My webMethods Server*
- *Administering webMethods OneData*
- *Administering webMethods Optimize* and *Configuring BAM*
- *Using BigMemory with webMethods Products* and BigMemory Max documentation
- Universal Messaging

See *About this Guide Online Information and Support* for information on accessing documentation.

Troubleshooting

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

- Look in log files.

  The table below lists the types of logs that record upgrade information and their locations.

Type of Log	Location
Installation and uninstallation	*new_Software AG_directory*/install/logs directory

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

- Go to the Knowledge Center on the Empower Product Support Website and read tech articles (for example, KB 1788395). Use keywords such as `upgrade` to search for relevant tech articles, and search for your old release, your new release, and all releases in between.
- Go to the Software AG TECHcommunity and:
  - Join the upgrade discussion forum (Forums > webMethods > Upgrade).
  - Read and share tips and tricks and other information in the upgrade wiki (Wiki > webMethods > Upgrade).

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

General Upgrade Procedure Requirements

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

- Perform the tasks in the upgrade documentation in the exact order in which they are presented. The task order is critical because your products have many inter-dependencies, including shared infrastructure and event driven architecture.
- 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.
- After you install the new products, you install the latest fixes. However, the upgrade procedure might take days or weeks to complete. Continue to check for and install new fixes regularly during the procedure.
- Do not start any new products before the instructions in this guide tell you to do so, or your database components could become corrupted.

API Gateway Requirements

Clustering Requirements

In API Gateway, a cluster is defined as multiple nodes where each node is an Integration Server instance that hosts API Gateway. Each Integration Server installation can contain multiple instances of API Gateway.

This section is intended for use with the upgrade procedure for this product as explained elsewhere in this guide. You must perform all documented tasks for your products in the order they are written in that procedure, with the additional tasks or exceptions noted below.

- When you install the new API Gateways, create a cluster of nodes that matches your old cluster. You can install the new API Gateways in parallel.

  If an old node includes multiple instances, create multiple instances in the new node. The new instance names do not have to be the same as the old instances names.

- When you install the fixes on the new API Gateways, you can install the fixes in parallel.
- When you start the data stores, start all the old data stores and then start all the new data stores.
- When you shut down the old products, shut down all cluster nodes.
- Migrate the data store for only one node. The data stores for the new nodes will be synchronized with the migrated data store when you start the new API Gateways after migration.
- Migrate old API Gateway configurations, data, and assets for every node.

Configure the cluster as instructed in *webMethods API Gateway Configuration Guide*.

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.

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.

Software AG Designer Requirements

If you are upgrading products involving Software AG Designer projects, make sure the target machine has enough space for the projects you will migrate (see "Migrate Software AG Designer and Business Process Data" on page 75).

Integration Server Requirements

General Requirements

Make sure the target machine has enough space for custom packages you will migrate. Custom packages include packages that were created by users in Software AG Designer and business process packages generated by users from Software AG Designer.

Clustering Requirements

In Integration Server, a cluster is defined as multiple Integration Server instances that point to the same ISInternal database component and connect to the same Cache Manager on a Terracotta Server Array. A cluster can include multiple Integration Server installations as well as multiple server instances within an Integration Server installation.

This section is intended for use with the upgrade procedure for this product as explained elsewhere in this guide. You must perform all documented tasks for your products in the order they are written in that procedure, with the additional tasks or exceptions noted below.

- When you install the new Integration Servers, create a set of installations that match your old cluster. You can install the new Integration Servers in parallel.
- When you install the fixes on the new Integration Servers, you can install the fixes in parallel.
- When you shut down the old products, shut down all cluster nodes.

Configure the cluster as instructed in *webMethods Integration Server Clustering Guide*.

Microservices Runtime Requirements

General Requirements

Make sure the target machine has enough space for custom packages you will migrate. Custom packages are packages that were created by users in Software AG Designer.

Clustering Requirements

In Microservices Runtime, a cluster is defined as multiple Microservices Runtime instances that point to the same ISInternal database component and connect to the same Cache Manager on a Terracotta Server Array. In pre-10.4 installations, a cluster could include multiple Microservices Runtime installations as well as multiple server instances within a Microservices Runtime installation. Starting in 10.4, Microservices Runtime supports only one server instance, so a cluster can include multiple Microservices Runtime installations only.

This section is intended for use with the upgrade procedure for this product as explained elsewhere in this guide. You must perform all documented tasks for your products in the order they are written in that procedure, with the additional tasks or exceptions noted below.

- When you install the new Microservices Runtimes, install a Microservices Runtime for each old server instance you want to migrate. You can install the new Microservices Runtimes in parallel.
- When you install the fixes on the new Microservices Runtimes, you can install the fixes in parallel.
- When you shut down the old products, shut down all cluster nodes.

Configure the cluster as instructed in *webMethods Integration Server Clustering Guide*.

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.

Do not redefine the environment variable after installing the new My webMethods Server.

My webMethods Server Clustering Requirements

In My webMethods Server, a cluster is defined as multiple My webMethods Server instances that point to the same MywebMethodsServer database component. A cluster can include separate My webMethods Server installations as well as server instances within a My webMethods Server installation.

This section is intended for use with the upgrade procedure for this product as explained elsewhere in this guide. You must perform all documented tasks for your products in the order they are written in that procedure, with the additional tasks or exceptions noted below.

- Install Universal Messaging before installing My webMethods Server.

  Note:
  Starting in 10.0, My webMethods Server clusters no longer use the server database to exchange JMS events or as a JMS provider. Instead Universal Messaging is required for synchronization among the cluster nodes.

- When you install the new My webMethods Servers, create a set of installations that matches your old cluster. You can install the new My webMethods Server installations in parallel.
- When you install the fixes on the new My webMethods Server installations, you can install the fixes in parallel.
- You do not need upgrade the Software AG Infrastructure; you can ignore those instructions in this guide.

- When you shut down the old products, shut down all cluster nodes.
- When you migrate server instances from old My webMethods Server installations to new My webMethods Server installations, you can migrate the installations in parallel.
- When you initialize the server instances, you can initialize them in parallel.

Configure the cluster as instructed in *Administering My webMethods Server*.

OneData Requirements

1. 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.

2. Go to the *old_Software AG_directory*/profiles/ODE/configuration/com.softwareag.platform.config.propsloader directory and open the com.softwareag.catalina.resource.pid-*repository_name schema*.properties files, where *repository_name* is the repository name you provided during installation and *schema* is md, wa, or ra. Note the database connection prefixes at the beginning of the names for the metadata, work area, and release area schemas. For example, in the metadata schema name jdbc/dev2md, the connection prefix is dev2. Also note any database parameters such as maxIdle, maxActive, and maxWait that are specified for the schemas.

3. Go to the *old_Software AG_directory*/profiles/ODE/bin/onedata/config directory. Open the repository.xml file and note the Repository Id (RepositoryID element) and Repository Name (Name element).

Universal Messaging Clustering Requirements

In Universal Messaging, a cluster is defined as multiple server instances in Universal Messaging installations that are configured for clustering.

This section is intended for use with the upgrade procedure for this product as explained elsewhere in this guide. You must perform all documented tasks for your products in the order they are written in that procedure, with the additional tasks or exceptions noted below.

- When you install the new Universal Messaging servers instances, create a cluster that matches your old cluster. You can install the new Universal Messaging server instances in parallel.
- When you install the latest fixes on the new Universal Messaging server instances, you can install the fixes in parallel.
- When you shut down the old products, shut down all cluster nodes.
- When you migrate server instances from old Universal Messaging installations to new Universal Messaging installations, you can migrate the instances in parallel.
- When you start the cluster nodes:

- If the new cluster nodes are on the same host machines as the old cluster nodes, start all new cluster nodes.
- If the new cluster nodes are on different machines than the old cluster nodes, follow the steps below. In those steps, If you are not using a remote join or a zone, "nodes" refers to nodes that are part of the cluster. If you are using a remote join or zone, "nodes" refers to nodes that are part of the remote join or zone.

    1. Go to the *new_Software AG_directory*/UniversalMessaging/server/*instance_name*/bin directory for one node and create a file named remote_realms_bootstrap.conf. Add a cross-host mapping property to the file for every node using the format *instance_name*.address=*new_host_name_or_IP_address* (for example, umserver1.address=myUM1.com or umserver2.address=192.168.0.1). Copy the file to the same location for all nodes.
    2. Start all nodes. The remote_realms_bootstrap.conf file in every installation will be renamed remote_realms_bootstrap_old.conf. When a source Universal Messaging server instance is bound to a specific network interface (absolute IP address), the interface of the migrated instance address will bind to 0.0.0.0 (all known interfaces on the port).
    3. Open Enterprise Manager and make sure the cluster is running correctly.

Configure the cluster as instructed in Universal Messaging documentation

3 Create an Upgraded Installation Using Command Central and Software Stacks (10.1 and Later)

You can upgrade Command Central software stacks using this procedure, or you can create stacks from existing product installations managed by Command Central and then upgrade them using this procedure.

1. Follow the steps in "Critical Factors and Requirements for Successful Upgrade" on page 15 and "Prepare the Old Environment for Migration" on page 34 in this guide.
2. Install the new release of Command Central and then open Command Central using instructions in *Software AG Command Central Help*).
3. If you are upgrading from 10.1 or 10.3, install the fixes below using instructions in *Software AG Command Central Help*.
   - Upgrade from 10.1: Install wmFix.SPM 17 and wMFix.SDRepository 9 on the Platform Manager of each installation you are upgrading. If the stack includes a database layer, install wMFix.DatabaseComponentConfigurator 3 on Database Component Configurator
   - Upgrade from 10.3: Install wmFix.SPM 12 on the Platform Manager of each installation you are upgrading. If the stack includes a database layer, install wMFix.DatabaseComponentConfigurator 6 on Database Component Configurator
4. If the installations you want to upgrade currently exist as standalone installations, create stacks from those installations. To do so, you will identify installations to Command Central, and Command Central will create one stack for each set of installations that are the same release, are installed on the same operating system and in a directory of the same name, and whose Platform Manager uses the same port and protocol (HTTP or HTTPS).

   Note:
   An installation can be part in only one stack. If an installation is already part of an existing, therefore, it will not be added to the new stack.

   a. Go to **Environments**, click the environment that contains the installations from which to create stacks, click  at the bottom right of the **Environments** pane, and then click **Auto-create stacks**.

   b. In the **Auto-create stacks** dialog box, in the **Product releases** drop down, click the release of the installations from which to create stacks or click **ALL**.

By default, Command Central will name the stacks 1, 2, 3, and so on. In the **Stack prefix** field, specify a prefix for those names that will help you better identify the stacks later.

c. Click **OK** to create the stacks.

d. If you need a database layer in a stack, add it manually.

e. If you do not want to include a particular layer in a stack, you can delete it.

5. Command Central needs its bootstrapper to install Platform Manager in the new product installation. Follow the instructions in your installation email from Software AG to download the bootstrapper of the new release for your operating system. Store the bootstrapper in the *Command Central_directory*/profiles/CCE/data/installers directory.

6. If you are going to install the new products on UNIX or Windows operating systems, make sure the target machines have a Secure Shell (SSH) server running and the system is configured for remote access with the user account for your products. For Windows:

   - If your Windows release has out-of-the-box SSH support, use the default SSH server on the Windows machine.
   - If your Windows release does not support SSH by default, set up OpenSSH on the Windows machine using a third-party tool such as Cygwin. For information about how to install Cygwin, go to https://cygwin.com/install.html or follow the instructions in the *Using Cygwin to Configure OpenSSH When Installing Platform Manager on a Remote Windows Machine* article located on Software AG TECHcommunity.

7. Create product and fix repositories for the new product release, import product license keys, and register any necessary credentials for the target machines using instructions in *Software AG Command Central Help*.

8. Depending on whether Command Central uses HTTP or HTTPS to connect to the Platform Manager installations from the infrastructure layer of the stack, ensure that the next port number or the one preceding it is free.

   - If Command Central uses HTTP to connects to a Platform Manager, make sure that this port number incremented by one is free. For example, if Command Central uses port 8092 to connect to a Platform Manager, ensure that port 8093 is free.
   - If Command Central uses HTTPS to connect to a Platform Manager, make sure that this port number decremented by one is free. For example, if Command Central uses port 8093 to connect to a Platform Manager, ensure that port 8092 is free.

   Note:

   You must keep this additional port free only during the stack upgrade.

9. Click **Stacks** at the top of the GUI. On the stack to upgrade, click ⋮, click **Upgrade stack**, click the type of upgrade to perform, and provide the requested values. You can now do one of the following:

   - Click **Dry run** at the bottom of the configuration page. Command Central will generate a composite template for each runtime component in the stack and validate the templates

(for example, by making sure the products in the templates exist in the repositories for the new release). You can edit the generated templates to reflect changes, deprecations, and removals as described above. You can then perform the upgrade using **Upgrade stack**, described below, or you can use the templates in automated upgrades.

- Click **Upgrade stack** at the bottom of the configuration page. Command Central will generate a composite template for each runtime component in the stack, validate the templates, and use the templates to perform the upgrade.

  If a product in a template does not exist in the repositories for the new release, the upgrade will fail and you will see a message that says you must edit that template. Click **Templates** at the top of the GUI; the generated templates will be named migration_generated_*stack_name*. Download the template and edit it with a text editing tool; for example, if the product no longer exists, you would delete the product entry in the template, or if the product has been replaced, you would change the product entry to the replacement product. When you are done editing, click  to import the template. Then click **Stacks** again, click , click **Upgrade stack** again, and choose to use existing templates.

10. Follow the steps in "Complete Final Upgrade Tasks for All Products" on page 137 in this guide.

4 Install New Products Using Command Central or Software AG Installer

Install New Products Using Software AG Installer

If you are using Software AG Installer, follow the instructions in *Using Software AG Installer*, *Installing Software AG Products*, and below to install your new products.

When running Software AG Installer:

1. For installation directory, specify a new Software AG installation directory.
2. Install the latest updates (see "Install Latest Updates on New Products" on page 31 for details).
3. On product panels or in response to prompts, provide the requested data as described below.

   Note:
   If you are installing the new products on the same machine as the old products, 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.

   The table below describes the data to provide on installer panels or in response to prompts for the indicated products.

Product	Guidelines
ActiveTransfer Server	Point the database connection at the ActiveTransfer database component you want to use with the new ActiveTransfer Server.
Apama	If installing the new Apama on the same machine as the old Apama, specify a new work directory. Software AG recommends including the release number in the work directory name, as shown in the default.
Integration Server	Point the database connection at the ISInternal database component you want to use with the new Integration Server.
Microservices Runtime	Point the database connection at the ISInternal database component you want to use with the new Microservices Runtime.
My webMethods Server	Select **Side-by-side installation for upgrade**. No instance will be created.
OneData	Enter the values you noted earlier from the old installation for ports, repository ID, repository name, and database connection prefixes and parameters. Make sure the database connections point to the OneData database components you want to use with the new OneData. **Note:** Reusing port values means that Web service clients of the old OneData will not have to change URLs to access the new OneData.
Trading Networks	Point the database connection at the TradingNetworks database component you want to use with the new Trading Networks.

Product	Guidelines
Universal Messaging	Select **Side-by-side installation for upgrade**. No instance will be created.

Install New Products Using Command Central

If you are using Command Central but are not using software stacks, follow the instructions in *Software AG Command Central Help* to install your new products, and also the latest updates (see "Install Latest Updates on New Products" on page 31).

Install Latest Updates on New Products

Install the latest updates on your new products while running the Software AG Installer or from Command Central. In addition:

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention MIG_*release*_MigrationFramework_Fix*number* and are listed under Common Library.
- Install fixes on database migration scripts. Database migration script fix names follow the convention *product* Database *release* Fix *number*.
- Upgrade from 9.9, 9.10, 9.12, or 10.1: If you installed a fix on the Software AG-provided JDK for the new release, and if you made changes (for example, security changes) to the JVM files in the *Software AG_directory*/jvm/jvm.bck directory in the old release , make the same changes to the JVM files in the *Software AG_directory*/jvm/jvm directory in the new release.

5 Prepare Old Products and Create ZIP Files for Cross-Host Migration

Shut Down Software AG Runtime and Disable Automatic Startup

Software AG Runtime starts automatically after installation. Shut down Software AG Runtime by stopping the Windows service or UNIX daemon. You can use the shutdown.{bat|sh} scripts in *new_Software AG_directory*/profiles/CTP/bin directory.

If you installed your new products on a Windows system, and you installed them as Windows services, the default startup mode for the services is Automatic. To prevent the new products from starting accidentally before this procedure instructs you to start them, set the services to Manual. 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 of the new products at this point. Do not start any of the new products before the instructions in this guide explicitly tell you to do so, or your database components could become corrupted.

Prepare the Old Environment for Migration

Install Fixes on Old Products

Install the latest product fixes on the old API Gateway, API Portal, CentraSite, Infrastructure Data Collector, My webMethods Server, Optimize, Process Engine, and Universal Messaging. After you install the fix on the old product, start the old product.

Upgrade from 9.9, 9.10, or 9.12 for ActiveTransfer: Install Fix 11.

Note:
The ActiveTransfer release number in those releases was 9.8.

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. Then 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
   ```

3. Run this command:

   ```
   acc> startall
   ```

4. Run this command to make sure all API Portal components are running:

   ```
   acc> list
   ```

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

```
acc> list tenants
```

6. 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.

   The table below explains how to back up each old tenant's data to a file for each old product release.

Old Release	Commands to Run
9.9 9.10 9.12	`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`
10.1 10.3 10.4 10.5	`acc> backup tenant tenant_name to full_path_to_backup_file username=your_user_name password=your_password`

7. 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

1. 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.

2. If the new and old CentraSites are on the same machine, make sure the old CentraSite is shut down.

3. 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 the appropriate command below.

   - For Windows, run `sbsExport.cmd` *full_path_to_ZIP_file*

- For UNIX, run `sbsExport.sh` *`full_path_to_ZIP_file`*

An example of this command for UNIX is as follows:

```
./sbsExport.sh /tmp/sbs_cs82_data.zip
```

4. If the old and new CentraSite installations are on different machines, copy the ZIP file to any directory on the machine that hosts the new CentraSite.

Prepare the Old Deployer

Start the old host Integration Server or Microservices Runtime. In the old Deployer, edit all connections to old source and target product servers to point to the new source and target product servers. You do not have to supply user names and passwords if you do not know them. For instructions, see *webMethods Deployer User's Guide*.

Prepare the Old Software AG Designer

Export Integration Server or Microservices Runtime 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 to file, go to **Window > Preferences > 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 view, right-click a governance project to export, click **Export**, and complete the wizard. The project folder will be exported.

   In the CloudStreams Providers view, right-click the provider project to export, click **Export**, and complete the wizard. The projects will be exported as an archive (ZIP file).

Export Preferences

If you want to migrate your preferences to the new Software AG Designer, you export them from the old Software AG Designer and then import them into the new Software AG Designer. 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 or Microservices Runtime

Start and Connect Products

1. Start the old Integration Server or Microservices Runtime and open the old Integration Server Administrator or Microservices Runtime Administrator.

2. If you are using Universal Messaging, make sure Integration Server or Microservices Runtime is connected to each Universal Messaging server that is acting as a webMethods Messaging provider. If you are using another JMS provider, make sure Integration Server or Microservices Runtime is connected to the JMS provider.

Verify Third-Party Library Compatibility with the New Release

If you have added any third-party (custom) jar files (for example, MySQL database driver jar files) to the *old_Software AG_directory*/IntegrationServer/lib/jars/custom or *old_Software AG_directory*/IntegrationServer/instances/**instance_name**/lib/jars/custom, make sure those jar files are compatible with new Integration Server or Microservices Runtime.

Suspend Triggers and Drain Queues

If you are not using Command Central to upgrade, perform the steps below.

1. Go to the **Settings > Quiesce** page. Click **Enter Quiesce mode** and set the time for the quiesce to occur to at least 1 minute, so Integration Server or Microservices Runtime has time to stop executing new incoming requests and to finish executing in-flight services. Make sure the Quiesce Report shows the status SUCCESS in every field. For instructions and details about specific actions that occur when Integration Server or Microservices Runtime is quiesced, see *webMethods Integration Server Administrator's Guide*.

2. Go to the **Settings > Messaging > webMethods Messaging Trigger Management** page. If the **Current Queue Counts** field does not show 0 for every trigger, diagnose and fix the problem (for example, the webMethods Messaging provider might not be active or might be slow to process requests from Integration Server or Microservices Runtime). Refresh the page until the **Current Queue Counts** field shows 0 for every trigger.

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

4. 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.

Prepare the Old MashZone NextGen

Upgrade from 9.10 or 9.12: MashZone NextGen 10.0 and later do not support the PPM Chart View. To remove the view from the MashZone NextGen database, open the MashZone NextGen Developer or MashZone NextGen Administrator, go to the API Console, and send this request:

```
{
  "version": "1.1",
  "sid": "AppService",
  "oid": "removeApp",
  "svcVersion": "0.1",
  "params": [
    "ppm-chart"
  ]
}
```

A response of `true` indicates that the view has been removed.

Prepare the Old My webMethods Server

When you 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 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 are 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.

   The table below lists the directories that contain the Kerberos-based SSO configurations and files to back up.

old_Software AG_directory/	**Files to Back Up**
profiles/ODE/configuration/tomcat/conf	server.xml
profiles/ODE/configuration	custom_wrapper.conf and jaas.config

old_Software AG_directory/	**Files to Back Up**
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

If you want to cluster Analytic Engines in the new installation, you will need a Terracotta Server Array. For planning information, see *Using BigMemory with webMethods Products*, *Configuring BAM*, and the BigMemory Max documentation.

Prepare the Old Zementis Predictive Analytics

1. Go to the *old_Software AG_directory*/Zementis/adapa-app/migration directory.
2. Back up Zementis Predictive Analytics models and resources by running this command:

   ```
   ./migration.sh backup Zementis_host_URL admin_user admin_password
   ```

   The backup is written to the adapa-backup-*yyyymmdd-hhmmss* directory, where *yyyymmdd-hhmmss* is the time you ran the command based on the UTC time zone.

Shut Down the Old Products

Shut Down Old Products Using Command Central

Shut down products from Command Central.

Shut Down Old Products Manually

The table below lists how to shut down specific products.

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`
	■ Stop the cloud agent by going to the *old_Software AG_directory*/API_Portal/server/bin directory and running this command:
	`CloudAgentApp.{bat|sh} stop`

Product	Shut Down Instructions
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.
BigMemory Max or Terracotta	Shut down all non-webMethods clients.

To shut down the other products on a Windows system:

- If the product is running as a service, shut down from the Windows Services window. Services are listed as Software AG *product release*.
- If the product is running as an application, shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > *product*** .

On a UNIX system, use the instructions in the product documentation for your old release.

Note:
It is especially important to shut down all old ActiveTransfer, Integration Server, Microservices Runtime, My webMethods Server, OneData, and Optimize instances that connect to database components you are going to migrate.

Create ZIP Files as Source of Old Product Installations 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 migration source.

Create ZIP File When Using Command Central to Migrate

If you are upgrading one environment using Command Central, you can create ZIP files of the old installations for supported products from Command Central or create them yourself as described in the next section.

If you are using composite templates to automate the upgrade, Command Central automatically creates a ZIP file of each old installation directory. The ZIP file includes all files except .log files. If you want to decrease the size of the ZIP file, see the table in the next section for the commands that will include only the required files for each product in the ZIP file.

Create ZIP File When Using Migration Utilities to Migrate

Create ZIP File for My webMethods Server

1. On the old machine, open a command window or shell, go to the *old_Software AG_directory*/MWS/bin/migrate or *old_Software AG_directory* /MWS/bin directory, depending on which old release you have, and run the command `ZIP-mws.{bat|sh}`. The command creates a ZIP file named mws.zip in *old_Software AG_directory*/MWS/bin/migrate directory.

2. Copy the mws.zip file to any directory on the machine that hosts the new 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 File for Optimize

1. On the old machine, open a command window or shell, go to the *old_Software AG_directory*/optimize/analysis/bin/migrate directory and run the command `ZIP-optimize.{bat|sh}`. The command creates a ZIP file named optimize.zip in the same directory. Copy the ZIP file to any directory on the machine that hosts the new Analytic Engine.

   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.

2. On the old machine, open a command window or shell, go to the *old_Software AG_directory*/optimize/dataCollector/bin/migrate directory and run the command `ZIP-dc.{bat|sh}`The command creates a ZIP file named dc.zip in the same directory. Copy the ZIP file to any directory on the machine that hosts the new Web Services Data Collector.

   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.

1. Go to the product's old machine and open a command window or shell.

2. Go to the Software AG directory that contains the old product and enter the command indicated in the table below. If multiple commands are listed, use the same ZIP file name in each command.

3. Copy the product's ZIP file to any directory on the machine that hosts the new 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.

The table below lists the command to run to create a ZIP file of the old product installation to use as the migration source.

Product	Command to Run
Application Platform	You can reduce the ZIP file size by first moving the log files out of the *old_Software AG_directory*/profiles/*product*/logs directory, where *product* is the host Integration Server instance, the host My webMethods Server instance, or SPM (for Platform Manager). `jar cfM ZIP_file_name.zip install/products profiles`

Digital Event Services	If your installation includes the file migrate.{bat|sh} in the *new_Software AG_directory*/common/migrate/DigitalEventServices/bin directory:
	`jar cfM ZIP_file_name.zip install/products`
	`jar ufM ZIP_file_name.zip common/DigitalEventServices`
	If the following directories exist, add them to the ZIP file:
	`jar ufM ZIP_file_name.zip profiles/product /configuration/DigitalEventServices`

Software AG Infrastructure (specifically, Software AG Runtime)	You can reduce the ZIP file size by first moving the log files out of the *old_Software AG_directory*/profiles/CTP/logs directory. `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	You can reduce the ZIP file size by first moving the log files out of the *old_Software AG_directory*/profiles/InfraDC/logs directory. `jar cfM ZIP_file_name.zip common/conf install/products profiles/InfraDC`
Integration Server	You can reduce the size of the ZIP file by first moving the log files out of the *old_Software AG_directory*/profiles/*instance_name*/logs, /IntegrationServer/instances/logs, and /IntegrationServer/instances/*instance_name*/logs directories. `jar cfM ZIP_file_name.zip *`
Microservices Runtime	For releases 10.3 and earlier, you can reduce the size of the ZIP file by first moving the log files out of the *old_Software AG_directory*/profiles/*instance_name*/logs, /IntegrationServer/instances/logs, and

Product	Command to Run
	/IntegrationServer/instances/*instance_name*/logs directories. For release 10.4 and later, you can reduce the size of the ZIP file by first moving the log files out of the /IntegrationServer/logs directories. `jar cfM ZIP_file_name.zip *`
MashZone NextGen	`jar cfM mzng.zip MashZoneNG`
OneData	`jar cfM ZIP_file_name.zip install/products profiles/ODE/bin/onedata/config/*`
Universal Messaging	`jar cfM ZIP_file_name.zip install/products UniversalMessaging/server`

6 Migrate Database Components

Cautions

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

Important:
The new release might require changes to the database components, such as new tables, columns, keys, or indexes. In this section, you run database migration 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. These changes might increase the size of your database.

Prepare Database Components for Migration

- The instructions in this guide assume you are using the same type of RDBMS in the new environment that you used in the old environment. If you want to use a different type of RDBMS in the new environment, there are special requirements that must be met before you can upgrade your products. Contact Software AG Professional Services for more information.
- Software AG strongly recommends using cloned databases when testing your upgrade. You can clone only those database components that you will want to use in your new environment. 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.
  - *Installing Software AG Products* describes the basic grants and privileges needed to work with product database components. Before cloning your 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.
- Some products offer features to archive or purge data from their database components. You can reduce the amount of time needed to migrate database components if you archive and purge them beforehand. For instructions on archiving or purging database components for Integration Server, Microservices Runtime, or Process Engine, see the Monitor product documentation. For instructions on archiving or purging database components for other products, such as Optimize and Trading Networks, see the product documentation.
- Make a backup of the product databases. Shut down all ActiveTransfer, Integration Server, Microservices Runtime, My webMethods Server, OneData, and Optimize instances that connect

to database components before making the backup. 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.

- If you are upgrading Trading Networks, and 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 database migration scripts in the next step. The database scripts are located in the *new_Software AG_directory*/common/db/TradingNetworks/TradingNetworks/scripts/*old_release_number*/*RDBMS* directory. If there is a conflict, drop the custom indexes.

Migrate Database Components Using Command Central

Use the Command Central command `sagcc exec administration` to migrate database components. The command syntax is shown below. The parameters are the parameters you use when running the Database Component Configurator and are documented in *Installing Software AG Products*.

```
sagcc exec administration product DCC_node_alias DatabaseComponentConfigurator
database migrate [parm1=value1 ] [parm2=value2 ]...
```

In the examples below, the database components are on a SQL Server RDBMS at jdbc:wm:sqlserver://DBserver:1433;databaseName=TESTDB, and the database user is webmuser with password webmpass.

- To upgrade Integration Server database components from 9.9 to the latest release:

```
sagcc exec administration product local DatabaseComponentConfigurator
database migrate db.type=sqlserver product=IS version=latest
db.username=webmuser db.password=webmpass db.name=TESTDB
db.url="jdbc:wm:sqlserver://DBserver:1433;databaseName=TESTDB"
```

- To upgrade the My webMethods Server database components to the latest release:

```
sagcc exec administration product local DatabaseComponentConfigurator
database migrate db.type=sqlserver component=MWS version=latest
db.username=webmuser db.password=webmpass db.name=TESTDB
db.url="jdbc:wm:sqlserver://DBserver:1433;databaseName=TESTDB"
```

To check the progress of the migration in Command Central, click **Jobs** on the Command Central web user interface.

Migrate Database Components Using Database Component Configurator

On the machine on which you installed the new Database Configuration, open a command window or shell, go to the *new_Software AG_directory*/common/db/bin directory, and migrate database components by running the command below.

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

Database scripts are executed as sections, with each section concluding with a commit to the database. If a migrate action fails (for example, because of a network outage, an expired connection, or invalid credentials), the failed section will not have been committed, and therefore can safely be re-attempted. To do so, re-enter the failed command and add the --resume option. You must specify the same values for all parameters except the credential parameters, which can be different if necessary.

After you run each command, check the log file log_*yyyymmddhhmmss* in the *new_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 a UNIX or Linux system, enclose the values " *`db_server_URL`* ," " *`existing_db_user`* ", and " *`password`* " in quotation marks.

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.

If you are upgrading...	Values for *product*...
ActiveTransfer Server	`-c {ActiveTransfer|ActiveTransferArchive}`
API Gateway	`-c APIGatewayEvents`
Archive database component (used to archive ISCoreAudit and ProcessAudit data)	`-c Archive`

Integration Server	`-c {ISInternal|ISCoreAudit|CrossReference|DocumentHistory|DistributedLocking}`
	If the listed database components are in the same schema, you can specify the following instead:
	`-pr IS`
	Note:
	Migrate the Integration Server database components before migrating other database components.

If you are upgrading...	Values for *product*...

Microservices Runtime	`-c {ISInternal|ISCoreAudit|CrossReference|DocumentHistory|DistributedLocking}`
	
	If the listed database components are in the same schema, you can specify the following instead:
	`-pr MSR`
	
	Note:
	Migrate the Microservices Runtime database components before migrating other database components.

My webMethods Server	`-c {MywebMethodsServer}`
	
	Or:
	`-pr MWS`
	
	Note:
	The CentralConfiguration database component is automatically migrated.

OneData	`-c {OneDataMetadata|OneDataReleaseArea|OneDataWorkArea}`

Optimize	`-c {Analysis|ProcessTracker|ProcessAudit|DataPurge|DatabaseManagement}`
	
	If the listed database components are in the same schema, you can specify the following instead:
	`-pr OPTI`
	
	Note:
	The OperationManagement database component is automatically migrated.

Process Engine	These database components contain business process run time data for in-progress, completed, or failed business process instances. Migrate these database components if you want to finish running in-progress process instances or resubmit completed or failed process instances from your old installation in your new installation.
	`-c {ProcessEngine|ProcessAudit}`
	
	If the listed database components are in the same schema, you can specify the following instead:
	`-pr PRE`

Rules Engine	`-c BusinessRules`

If you are upgrading...	Values for *product*...
Reporting and Staging database components (used to simulate business processes)	`-c {Reporting|Staging}`

Trading Networks	`-c {TradingNetworks|TradingNetworksArchive}`
	
	If the listed database components are in the same schema, you can specify the following instead:
	`-pr TN`

7 Migrate Software AG Infrastructure

Software AG Infrastructure Configurations, Data, and Assets that Will be Migrated

- Software AG Runtime audit, debug, JAAS, JMX, port, security, single-sign on, watchdog, and Web Services Stack configurations.
- User repository and password store.
- Log4j 2 settings, as follows:
  - Upgrade from 10.4 and earlier: The utility migrates any custom non-default loggers and the org.apache.log4j.RollingFileAppender and org.apache.log4j.ConsoleAppender appenders from the old log_config.xml file to the new log4j2.properties file.
  - Upgrade from 10.5 and later: The utility migrates the entire contents of the old log4j.properties file to the new log4j2.properties file.
- Java Service Wrapper customizations you made in the old custom_wrapper.conf files, including #include directives and comments (but not associated properties).
- Users, groups, and roles.
- Proxy configuration files for proxy types HTTP, HTTPS, FTP, and SOCKS. The utility migrates files for proxies that are not already configured in the new installation.
- Starting in 10.0, Software AG Security Infrastructure no longer supports SSX or SSX LoginModules. The migration utility migrates LoginContexts that do not contain SSX LoginModules. However, if an old LoginContext contains SSX LoginModules, the migration utility does the following:
  - If LoginContext exists in old and new release, the utility adds a comment to the new LoginContext that no migration was performed.
  - If LoginContext exists in old release but not in new release, the utility adds an empty LoginContext to the new release with comment that no migration was performed. This is to prevent software that used the old LoginContext from failing because of missing context.

Starting in 10.1, the migration utility does not support migration of JKS files that are not in the Software AG installation directory.

Migrate Software AG Infrastructure

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

1. Make sure none of the new products are running. If the new and old products are on the same machine, make sure the old products are shut down.

2. Run the Software AG Infrastructure migration utility as described in "Migrate Products Using Migration Utilities" on page 121.

Complete the Software AG Infrastructure Upgrade

#include directives are migrated to the end of the new custom_wrapper.conf file; check them and adjust as necessary. The custom_wrapper.conf file is located in the *Software AG_directory*/profiles/CTP/configuration directory.

Note:
As noted in the *Software AG Infrastructure Administrator's Guide*, you should never modify the wrapper.conf file unless instructed to do so by Software AG. If you did so, however, manually copy values for properties you modified in the old wrapper.conf file to the corresponding properties in the new custom_wrapper.conf file.

8 Migrate BigMemory Max and Terracotta

Migrate BigMemory Max

Note:
Terracotta BigMemory Max was renamed BigMemory Max in release 10.5.

This section explains how to migrateBigMemory Max clients and servers that are deployed in support of webMethods products. For all other setups, see the BigMemory Max documentation.

Follow the instructions in *Using BigMemory with webMethods Products* to install the 4.1.4 or later license key and tc-config.xml file.

You must upgrade all BigMemory Max clients and servers in the cluster to the new release before restarting the cluster.

If you have a single server:

1. Shut down your old Terracotta Server Array.
2. Start the new Terracotta Server Array.

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 the new active server.
4. Start the new mirror server.

Migrate Terracotta

Note:
Terracotta DB was renamed Terracotta in release 10.5.

Terracotta is significantly different from BigMemory Max because Terracotta is designed for storage and caching of typed data, while BigMemory Max is designed for caching of opaque data. Therefore, you cannot directly migrate data and configuration for a BigMemory Max server to a Terracotta server. If you want to migrate, you will have to create custom tooling that takes your particular usage into account.

Terracotta can reuse data stored on disk from earlier 10.x releases, as described below.

Migrate Terracotta Management Console Configuration and Data

1. Go to the *old_Software AG_directory*/TerracottaDB/tools/management/conf directory and copy the tmc.properties file over the same file in the new installation.

2. Go to the *old_Software AG_directory*/TerracottaDB/tools/management directory and copy the data directory to the same location in the new installation.

3. Start Terracotta Management Console.

4. If you receive error messages, open the tmc.properties file and make sure all the properties are set correctly. For instructions, see the product documentation.

Migrate Ehcache 3.x Data

If you have Ehcache 3.x data in restartable caches contained in a restartable cache manager, you can migrate that data to a new cluster without moving the platform data roots. Since the data migration works at the data directory level, data for all restartable cache managers that use the same data directory will be migrated together.

1. Shut down the old cluster.

2. Start the active servers in the new cluster with the same number of stripes as the old cluster.

3. Create the cache manager configuration using a client. The cluster URI, including the cluster tier manager name for the cache manager, can be different from the URI for the old cluster. If the name part of the URI is different, specify the old name as the restart identifier when using the cache manager configuration API, so the system can map the data corresponding to a given cache manager correctly. If there is more than one cache managers under the same data directory, use the configuration API to create all the cache managers in the target cluster. For instructions, see the section on fast restartability in the *Ehcache API Developer Guide*.

4. Shut down the new cluster.

5. For all restartable cache managers on all servers you want to migrate, copy the contents of the Ehcache directories (caching data roots) from the data directories of all active servers in the old cluster to the data directories in the new cluster. The data directory paths on the new cluster can be different from those on the old cluster.

6. Start the new cluster. Terracotta will load all the cache data that was migrated from the old cluster.

9 Migrate Universal Messaging

Universal Messaging Configurations, Data, and Assets that Can be Migrated

- Server instances.
- Server instance data directories.

Other Actions Performed as Part of the Migration

Universal Messaging 10.7 does not support the use of a JMS engine with queues. If your old release included queues configured to use a JMS engine, Universal Messaging automatically reconfigures those queues to use the default queue engine when you first start the Universal Messaging 10.7 server.

Migrate Universal Messaging

1. If you are upgrading a Universal Messaging cluster, read the clustering instructions in the Universal Messaging documentation.

2. If the new and old Universal Messaging installations are on the same machine, make sure the old installation is shut down.

3. You can migrate using either of two methods.

   - To migrate using the migration utility, follow the instructions in "Migrate Products Using Migration Utilities" on page 121.
   - To migrate using Command Central, use the Command Central command below. If you want to use a custom migration, first run the migration utility manually to create the migrate.dat file (see "Migrate Products Using Migration Utilities" on page 121 for instructions), then specify the migrate.dat file on the `importFile` parameter. If you want to migrate all instances, do not specify the `instanceName` option.

     ```
     sagcc exec administration product case_sensitive_target_node_alias
     NUMRealmServer migration migrate
     {srcDir|srcFile}=full_path_to_{old_Software AG_directory|ZIP file}
     [importFile=full_path_to_migrate.dat]
     [instanceName=name[,name...]]
     ```

     If you installed Template Applications and Enterprise Manager in the new installation, the migration creates Template Applications and Enterprise Manager instances.

Complete the Universal Messaging Upgrade

1. In release 9.8, the nserver.conf and nserverdaemon.conf files were combined into a single file named Server_Common.conf. The Server_Common.conf file is stored in the *new_Software AG_directory*/UniversalMessaging/server/umserver/bin directory.

- Upgrade from 9.9 or 9.10: If you made any custom changes to the Server_Common.conf file for an old instance, make the same changes in the Custom_Server_Common.conf file for the corresponding new instance.
- Upgrade from 9.12 or 10.1: If you made any custom changes to the Server_Common.conf or Custom_Server_Common.conf file for an old instance, make the same changes in the Custom_Server_Common.conf file for the corresponding new instance.

2. If the data directory for an old instance was not in the default location specified in the Software AG Installer, the migration utility did not migrate the data directory. The default location for the data directory is the *new_Software AG_directory*/UniversalMessaging/server/*instance_name*/data directory. You can use the old data directory in the old location or copy it to a new location. If you copy it, also do the following:
   - Edit the DDATADIR parameter in the Server_Common.conf file for the new instance to point to the new data directory location.
   - Go to the *new_Software AG_directory*/UniversalMessaging/server*instance_name*/bin directory, open the nstopserver.{bat|sh} script in a text editor, and edit it to point to the new data directory location.
3. If you modified the old default JAAS configuration file, go to the *old_Software AG_directory*/UniversalMessaging/server/*server_name*/bin directory, open the jaas.conf file, and copy the changes to the same file in the new installation. Also copy references to external resources such as truststore files and then modify the references as necessary.

10 Migrate My webMethods Server

My webMethods Server Configurations, Data, and Assets that Can be Migrated

- Server instances.
- JAAS configuration files.
- If the old My webMethods Server hosted Business Console, the user preferences, AppSpaces, and gadgets for Business Console.
- Java Service Wrapper customizations you made in the old custom_wrapper.conf files, including #include directives and comments (but not associated properties).

Other Actions Performed as Part of the Migration

- Old configuration files are deleted, and old properties that are not used by the new My webMethods Server are deleted from the new configuration files.
- The My webMethods Server installation directory is synchronized with the database.
- If you specified a URL for Universal Messaging, the existing JNDI URL parameter is updated or a new JNDI URL parameter is added to the cluster.xml file, as appropriate.

Embedded Database No Longer Supported

Starting in My webMethods Server 10.3, My webMethods Server no longer offers an embedded database, so the My webMethods Server migration utility will not migrate instances that used an embedded database. Instead you can create a new instance, and then transfer applications and content from the old instance to the new instance using Deployer or the content import/export functionality of My webMethods Server. For instructions, see *webMethods Deployer User's Guide* or *Administering My webMethods Server*, respectively.

Migrate My webMethods Server

1. If you are upgrading a My webMethods Server cluster, read " My webMethods Server Requirements" on page 22.

2. If the new and old My webMethods Servers are on the same machine, make sure the old My webMethods Server is shut down.

3. You can migrate using either of two methods.

   - To migrate using the migration utility, follow the instructions in "Migrate Products Using Migration Utilities" on page 121.
   - To migrate using Command Central, use the Command Central command below. If you want to use a custom migration, first run the migration utility manually to create the migrate.dat file (see "Migrate Products Using Migration Utilities" on page 121 for

instructions), and then specify the migrate.dat file on the `importFile` parameter. If you want to migrate all instances, do not specify the `instanceName` parameter.

```
sagcc exec administration product target_node_alias
MwsProgramFiles migration migrate
{srcDir|srcFile}=full_path_to_{old_Software AG_directory|ZIP file}
[importFile=full_path_to_migrate.dat]
[instanceName=name[,name...]]
[newNodeName=instance_name:new_node_name[,instance_name:new_node_name...]]
[cloneDbURL=URL cloneDbUser=user cloneDbPassword=password]
```

Note:
If you migrate the instances, and then have to re-run the migration later for some reason, Command Central handles the re-migration differently than the migration utility. If migrated instances in the new installation have the same name as old instances, the utility deletes the migrated instances and then re-migrates the old instances. Command Central, conversely, skips the migrated instances and only re-migrates old instances that do not yet exist in the new installation.

Complete the My webMethods Server Upgrade

Check Java Service Wrapper #include Directives

The migration utility migrates #include directives to the end of the new custom_wrapper.conf file; check them and adjust as necessary. The custom_wrapper.conf file is located in the *Software AG_directory*/profiles/MWS_*instance_name*/configuration directory.

Note:
As noted in the *Software AG Infrastructure Administrator's Guide*, you should never modify the wrapper.conf file unless instructed to do so by Software AG. If you did so, however, manually copy values for properties you modified in the old wrapper.conf file to the corresponding properties in the new custom_wrapper.conf file.

Initialize a My webMethods Server Instance

The initial startup of the instance might a while to complete (in rare cases, an hour or more).

1. Initialize the new My webMethods Server server instance by going to the *new_Software AG_directory*/MWS/bin directory and running this command:

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

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

2. 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 one of the new My webMethods Server instances 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 instance.

Switch from Clone to Live Database

If you used a clone database to test your new installation (recommended), and you later want to use your live database with the new installation you tested, you must run the My webMethods Server migration utility again to make the switch. The utility synchronizes the My webMethods Server installation directory with the database.

1. Run the My webMethods Server migration utility again.
2. Migrate each server instance again. When the utility asks for the database to use with each instance, choose the live database.
3. When the migration utility asks whether to delete or keep the migrated instances in the new installation, choose to delete the migrated instances.

Complete Business Rules Upgrade

Business Rules projects currently deployed to My webMethods Server are migrated when you first start a server instance. In a cluster, the first cluster node to start up migrates all rule projects. During this migration, all rule projects deployed on My webMethods Server are unavailable. The My webMethods Server log file contains details about the rule project migration. By default, the log file is stored in the *Software AG_directory*/MWS/server/*instance_name*/logs directory.

1. If the new installation directory has a different name than the old installation directory, check the validity of the file system location used by Deployer for Business Rules asset deployment, and restart My webMethods Server if updates are necessary. For instructions, see *Working with Business Rules in My webMethods*, chapter "Hot Deploying and Merging Rule Projects with webMethods Deployer" and section "Configuring My webMethods Server."

2. If you use business verification or data providers, log on to one of the new My webMethods Server instances as Administrator and go to the **Administration > My webMethods > System Settings > webMethods Business Rules Settings** page. Update the settings as necessary and save the changes.

   For more information, see *Working with Business Rules in My webMethods*. For business verification, see chapter "Rule Verification Overview" and section "Configuring a Server Connection for a PreconfiguredVerification." For data providers, see chapter "Working with Decision Tables" and section "Configuring a Server Connection for a Pre-configured Data Provider Service."

11 Migrate Integration Server, Microservices Runtime, and Hosted Wm Packages

Packages

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

- Custom packages. These include the following:
  - Integration Server or Microservices Runtime packages created by users in Software AG Designer.
  - Business process packages generated by Integration Server 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 or Microservices Runtime on the product selection tree. On the tree, they are listed using their product names. However, in the file system and within Integration Server or Microservices Runtime, 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), Process Engine (WmPRT), Trading Networks (WmTN), adapters, and eStandards Modules.

Configurations, Data, and Assets that Can be Migrated

Integration Server

- Custom (user-created) packages.
- Password store, and passwords.
- Configuration files for Integration Server, hosted products such as Trading Networks, and hosted Wm packages that are also hosted on the new Integration Server.
- JDBC connection pool configurations, database drivers, and functional aliases.
- Starting in 9.5, custom (user-created) jar files.
- JAAS configuration files.
- Java Service Wrapper customizations you made in the old custom_wrapper.conf files, including #include directives and comments (but not associated properties).

Microservices Runtime

- Custom (user-created) packages.
- Password store, and passwords.
- Configuration files forMicroservices Runtime, hosted products such as CloudStreams, and hosted Wm packages that are also hosted on the new Microservices Runtime.

- JDBC connection pool configurations, database drivers, and functional aliases.
- Custom (user-created) jar files.

Other Actions Performed as Part of the Migration

- Old configuration files are deleted, and old properties that are not used by the new installation are deleted from the new configuration files.
- If the old installation used the embedded database, the database tables from the old installation are copied to the new installation and the tables are upgraded to the new format, if the format has changed.
- The SERVER ID stored in database tables and configuration files is updated to reflect the new host machine.
- A new property named "Validate schemas using Xerces" is added to existing Web service descriptors, and the new property is set to the same value to which the watt.server.wsdl.validateWSDLSchemaUsingXerces parameter was set in the old installation. The new property replaces the functionality provided by that parameter. For information about the new property, see *webMethods Service Development Help*.
- Keystore aliases of type PKCS12 whose keystore provider name is SunJSSE are updated to use Bouncy Castle as a provider.
- If you are migrating Business Rules data, Business Rule projects are upgraded. You might see XML parsing messages due to a Java bug; you can ignore these messages.
- If you are upgrading CloudStreams, configuration artifacts related to administering CloudStreams Server are migrated.

Migrate Integration Server or Microservices Runtime

1. If you are upgrading a cluster, read " Integration Server Requirements" on page 21 or "Microservices Runtime Requirements" on page 21.

2. If you are upgrading an Integration Server or Microservices Runtime that hosts Deployer, follow the instructions in "Migrate the Asset Build Environment, Deployer, and Application Platform" on page 89.

3. If the new and old installations are on the same machine, make sure the old installation is shut down.

4. The migration 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 *new_Software AG_directory*/IntegrationServer/bin/migrate directory, open the packages.cnf file, and add a <value name><\value> tag that identifies those custom packages (for example, <value name="WmFINMessages">WmFINMessages</value>).

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

5. Migrate Microservices Runtime using the Integration Server migration utility. Follow the instructions in "Migrate Products Using Migration Utilities" on page 121.

   Migrate Integration Server using either of two methods.

   - To migrate using the Integration Server migration utility, follow the instructions in "Migrate Products Using Migration Utilities" on page 121.
   - To migrate using Command Central, use the Command Central command below. If you want to use a custom migration, first run the migration utility manually to create the migrate.dat file (see "Migrate Products Using Migration Utilities" on page 121 for instructions), and then specify the migrate.dat file on the `importFile` parameter. If you want to migrate all instances, do not specify the `instanceName` parameter.

   ```
   sagcc exec administration product target_node_alias
   integrationServer migration migrate
   {srcDir|srcFile}=full_path_to_{old_Software AG_directory|ZIP file}
   [importFile=full_path_to_migrate.dat]
   [instanceName=name [newInstanceName=name]]
   [cloneDbURL=URL cloneDbUser=user cloneDbPassword=password]
   ```

Complete the Integration Server Upgrade

Check Java Service Wrapper #include Directives

Any Java Service Wrapper #include directives are migrated to the end of the new custom_wrapper.conf file; check them and adjust as necessary. The custom_wrapper.conf file is located in the *Software AG_directory*/profiles/IS_*instance_name*/configuration directory.

Note:
As noted in the *Software AG Infrastructure Administrator's Guide*, you should never modify the wrapper.conf file unless instructed to do so by Software AG. If you did so, however, manually copy values for properties you modified in the old wrapper.conf file to the corresponding properties in the new custom_wrapper.conf file.

Complete the Integration Server or Microservices Runtime Upgrade

Customize Startup Files

For Integration Server, the 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 the new installation.

If you are upgrading from Microservices Runtime 10.4 or earlier, copy any changes you made in the Java Service Wrapper to the startup scripts. For instructions, see *webMethods Integration Server Administrator's Guide*.

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 the new Integration Server or Microservices Runtime.

2. Review the migrationLog.txt file in the *new_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 the new Integration Server Administrator or Microservices Runtime Administrator, and point to the new Integration Server or Microservices Runtime.

   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.

Upgrade from 10.1: Save and Synchronize an Updated Publishable Document Type

If your solution includes publishable document types with an encoding type of protocol buffers, you need to edit, save, and synchronize the publishable document types to Universal Messaging while the queues of subscribing triggers are empty.

Connect Integration Server or Microservices Runtime and Hosted Products to Database Components

If you did not migrate functional aliases, database driver configurations, and JDBC connection pool configurations, connect Integration Server or Microservices Runtime and the products it hosts to their database components. For instructions, see *Installing Software AG Products*.

Complete the Adapters or eStandards Modules Upgrade

See the adapter or eStandards Module product documentation.

Complete the ActiveTransfer Upgrade

If you used custom keystore files in the old ActiveTransfer installation, make sure they are available and accessible in the location specified in the new installation.

If you have active file shares and did not configure the mft.sharing.account.tempdir property in the old installation, go to the *new_Software AG_directory*/IntegrationServer/instances/*instance_name*/packages/WmMFT/config directory, open the properties.cnf file, and specify the old TempAccounts path on the mft.sharing.account.tempdir property.

When you start ActiveTransfer, it will modify listener (port) names to comply with Command Central standards. If ActiveTransfer finds duplicate listener (port) names, it will add an underscore and a unique number to each name. If a listener (port) name has a white space, it will replace that white space with an underscore.

Complete the CloudStreams Upgrade

Back up any custom packages that contain CloudStreams artifacts, and then 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 the new CloudStreams. Artifacts include SOAP and REST connector services, connector listeners, and connections. The utility logs the results of migration to the Integration Server or Microservices Runtime server log.

The table below provides the input signature for the `pub.cloudstreams.migration:migrate` service.

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.
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 table below lists the fields in each record in the array for each custom package specified in the input signature for the `pub.cloudstreams.migration:migrate` service.

Field	Description
packageName	**String**. Name of the custom package.

Field	Description
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.

Configure TLS 1.1 and 1.2 Trust Store

For SaaS back ends that have certificates that are not part of the JVM trust store, create and apply a new JKS trust store in Integration Server or Microservices Runtime with the certificates of the SaaS back end. The trust store should contain all certificates in the certificate chain. This is a standard practice for setting up secure exchange of certificates. If the trust store is not present in such cases, certificate related errors will occur.

1. In Integration Server Administrator or Microservices Runtime Administrator, go to the **Security > Keystore** page and create the trust store. For detailed instructions, see the section on securing communications in *webMethods Integration Server Administrator's Guide*.

   Note:
   You can also create the trust store using a publicly available tool.

2. Apply the trust store in the connection's advanced property **Trust Store Alias**. For detailed instructions, see the CloudStreams documentation for the Provider you are using.

12 Migrate Software AG Designer and Business Process Data

Migrate Software AG Designer

Before Migrating

1. Open the new Software AG Designer and point to a new workspace. For example, you can accept the default workspace*release* (for example, workspace107).

2. Install any third-party features you need (for example, support for Subversion). The Eclipse release installed with the new Software AG Designer is Eclipse 4.15, so make sure any features you add are compatible with that Eclipse release.

3. 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 new 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, then review the messages on the Error tab to see whether any other paths need to be updated.

   c. Restart Software AG Designer.

Migrate Apama, Application Platform, Business Process, and Business Rule Projects

1. If you are migrating Apama, copy any files you modified from the old work directory to the new work directory.

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. If you are migrating Apama, use **Project** > **Clean** to force a rebuild of generated project artifacts under the control of Software AG Designer.

4. Verify the projects.

   The table below explains how to verify the projects for the listed products.

Product	Verify
Apama	All projects build without errors. If errors appear in the **Problems** tab, click each error and resolve it.
Application Platform	All 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.
Business Process	All imported business process projects appear in the **Solutions** tab under the Processes node.
Business Rules	All imported business rules projects appear in the **Solutions** tab under the Rules node.

5. If you imported Business Rule projects, right-click each project in the Rules Explorer and then click **Upgrade Project**.

6. If you are migrating Apama:

   a. Re-run any build or deployment scripts to make sure artifacts generated from your projects, such as Apama queries and plug-ins, are up to date. This task could include re-exporting and re-running Ant scripts, executing engine_deploy, rebuilding Docker images, and re-running custom scripts. Review the output for errors and warnings.

   b. If you use Apama dashboard deployments, create new deployment packages and install the deployment packages.

   c. If any project imports event types from Digital Event Services, you might need to re-import those event types. Open the project, open the EventTypeList file under config/connectivity/DigitalEventServices, and click **Sync from Digital Event Services**. If you see changes, save the file.

d. If any project includes a custom configuration for Digital Event Services, you will see a /DigitalEventServices directory. Erase the custom configuration and re-create it from scratch using Digital Event Services tooling or Command Central.

Migrate Task Application Projects

1. Update your preferences.

   a. Go to **Window > Preferences > Server > Runtime Environments**. If the **Installed server runtimes** list does not include a My webMethods Server from the new release, add one.

   b. 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**.

2. Import your projects.

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

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

   c. 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.

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

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

4. 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.

5. Publish the migrated task application projects to a My webMethods Server from the new release. For instructions, see *webMethods BPM Task Development Help*.

Redo Attachment List Control Customizations in CAF Applications

In 9.10, the CAF **Attachments List** control was improved; for example, the Java applet-based drag and drop panel was replaced by an HTML5-based equivalent.

Your CAF applications were migrated when you ran the My webMethods Server migration utility, but any customizations you made to the **Attachments List** control in your CAF applications were not. Manually redo the customizations in the migrated CAF applications.

Migrate CloudStreams Server Definitions and Projects

1. Go to **Window > Preferences > Software AG > CloudStreams > Servers**. Click **Import**, select the .properties file you exported earlier, and click **Open**. Software AG Designer asks whether to overwrite existing server definitions with the imported definitions. Click **OK** twice.

2. Open the CloudStreams Development perspective.

3. In the CloudStreams Governance view, right-click empty space and click **Import** to open the wizard. In the **Select Root Directory** field, identify the directory that contains the governance projects you exported earlier. In the **Projects** box, select the governance projects to import from the exported project folder. Click **Copy projects into workspace** and then click **Finish**. Click the **Governance** tab and make sure all imported governance projects appear.

4. In the CloudStreams Providers view, right-click empty space. If you want to import provider projects you exported to an archive (ZIP file) earlier, click **Import Archive File**. If you want to import provider projects from another CloudStreams, click **Import from Workspace**.

Migrate Integration Server or Microservices Runtime Definitions

1. 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.

Migrate Mobile Projects

1. Go to **Window > Preferences > Software AG > Mobile Development** and update the Mobile Designer installation directory to the MobileDesigner subdirectory in your new Software AG Designer installation directory.

2. Import your projects.

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

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

c. In the Import Projects panel, click **Select Root Directory** and go to the workspace that contains your mobile projects. In the **Projects** box, select the projects to import.

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

3. In Package Explorer, follow the steps below. For instructions on resolving issues and errors, see *Migrating Mobile Projects* in *webMethods Mobile Development Help*.

a. Go to the model directory for each imported project and double-click the .aml file. Check for warnings and errors and resolve any issues.

b. Right-click the root node for each imported project and then click **Generate Source Code > Application Model and API**. Resolve any compile errors.

c. If you are using Mobile Administrator (instead of the Jenkins-based Remote Multi-Build), enable the Mobile Administrator Remote-Multi-Build ant target. Go to the root of your project, open the build.xml file, and add the text below to the end of the file as a single line:

```
<import file="${env.MOBILE_DESIGNER}/plugins/MobileAdministrator/
v1.0.0/targets.xml"/>
```

4. If you later have problems with Android handset targets, such as problems running your project in the Phoney simulator, go to each project's /target directory in Package Explorer and remove all existing handset targets. Then re-add the targets for the project by running the ++Activate-Handset ant target.

Complete the Business Process Upgrade

1. If you have running process instances that are based on a process model created before you upgraded, and you now want to regenerate that process model in the new Software AG Designer, 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.

2. If you migrated business process packages that were generated by users from Software AG Designer, make sure the packages exist in the *new_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 new Integration Servers that host Process Engines. 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*.

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

5. If your old and new installations are on different machines and you want to be able to resubmit processes and process steps that ran before you upgraded, follow the steps below.

   If the new Integration Server that hosts Monitor is connected to a messaging product and has process model fragments, select the **Resubmit to local IS** check box, click **Submit**, and then reload the package.

   If the new Integration Server that hosts Monitor is not connected to any messaging product and has no process model fragments, clear the **Resubmit to local IS** check box, click **Submit**, and then reload the package.

6. Go to the **Settings > Messaging > webMethods Messaging Trigger Management** page and make sure document retrieval is enabled for all webMethods Messaging triggers on new Integration Servers. For instructions, see *webMethods Integration Server Administrator's Guide*.

13 Migrate Infrastructure Data Collector and Optimize

Configure and Start the Terracotta Server Array

If you are clustering Analytic Engines in the new release, you must configure and start the Terracotta Server Array. For instructions, see *Using BigMemory with webMethods Products*, *Configuring BAM*, and the BigMemory Max documentation.

Update Connections to Other Products

Update Connection to Process Engine

If you are using Optimize for Process, follow the steps below.

1. Open Integration Server Administrator and point to a new Integration Server 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 Universal Messaging host name and port as nsp://*host*:*port*. The default port is 9000.

5. Click **Submit** and then reload the WmPRT package.

6. Repeat these steps for every new Integration Server that hosts a Process Engine.

Update Connection to Optimize Support Package

If you are using Optimize built-in services, follow the steps below.

1. Open Integration Server Administrator or Microservices Runtime Administrator and point to a new Integration Server or Microservices Runtime 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 Universal Messaging host name and port as 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 or Microservices Runtime that hosts an Optimize Support package.

Migrate Infrastructure Data Collector

Infrastructure Data Collector Configurations, Data, and Assets that Can be Migrated

- Asset configuration files.
- SNMP, JAAS, JMX, SSH, security, log, platform debug, Web Services Stack, and watchdog configurations.
- Java Service Wrapper customizations you made in the old custom_wrapper.conf files, including #include directives and comments (but not associated properties).

Migrate Infrastructure Data Collector

1. If the new and old Infrastructure Data Collectors are on the same machine, make sure the old Infrastructure Data Collector is shut down.

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

3. Run the Infrastructure Data Collector migration utility as described in "Migrate Products Using Migration Utilities" on page 121.

Complete the Infrastructure Data Collector Upgrade

1. Any #include directives are migrated to the end of the new custom_wrapper.conf file; check them and adjust as necessary. The custom_wrapper.conf file is located in the *Software AG_directory*/profiles/InfraDC/configuration directory.

   Note:
   As noted in the *Software AG Infrastructure Administrator's Guide*, you should never modify the wrapper.conf file unless instructed to do so by Software AG. If you mistakenly modified the old wrapper.conf file, manually copy values for properties you modified in the old wrapper.conf file to the corresponding properties in the new custom_wrapper.conf file.

2. Rediscover your assets. Open the new My webMethods Server, go to the **Applications > Administration > Analytics > Infrastructure Components > Discovery** page, and click **Run Discovery**.

Migrate Optimize

Optimize Configurations, Data, and Assets that Can be Migrated

- Analytic Engine, database, data maintenance, event publication, JMS EventAction, mail, monitoring, process tracking, SNMP alert, station, and WS action settings.
- JNDI and journal logging configurations.

Migrate Optimize

1. If the new and old Analytic Engines are on the same machine, make sure the old Analytic Engine is shut down.

2. Run the Optimize migration utility as described in "Migrate Products Using Migration Utilities" on page 121.

Deploy the Optimize Environment

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

Not Reuse the Old Optimize Central Configurator System (CCS) Environment Definition

Start the new My webMethods Server, open My webMethods, and configure your new Optimize 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 new 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.

Go to *Deploy the New Optimize Environment*, below.

Reuse the Old Optimize CCS Environment Definition

1. Start the new My webMethods Server and open My webMethods.

2. Migrate the environment to use in the new installation. Go to the **Applications > Administration > System-Wide > Environments > Define Environments** page, select the environment, and click **Migrate Environment**.

The Configured and Ready to Deploy columns should show green check marks. If the Ready to Deploy column shows a red circle instead, click the environment, and then click the **Validate** tab.

3. If your old installation contained other environments, and you want to migrate those as well, repeat the previous step. These additional environments will be migrated in their current state (for example, if they are unconfigured in the old installation, they will remain unconfigured in the new installation).

4. If the database name or connection pool settings differ from those in the old installation, go to the **Applications > Administration > System-Wide > Environments > Database Pool Configuration** page and update as necessary. In the **Database Connection** area, click **Test** and make sure the message `Test Passed` displays at the bottom of the page. Click **Save**.

5. Click the **Define Environments** tab, click the environment to deploy, modify the configuration if necessary, and then click **Finish**.

Deploy the New Optimize Environment

1. Start the new Optimize components you installed. These components can include the Analytic Engine, Infrastructure Data Collector, and the Web Services Data Collector.

2. Start the Universal Messaging you are using with the new release.

3. Deploy your new Optimize environment. For instructions, see *Configuring BAM*.

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 new Analytic Engine and the new Integration Server that hosts Monitor. For the hosts, you can specify 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 new Analytic Engine. For the host, you can specify DNS name or IP address.

5. Start the Analytic Engine and then click **Check Server Status** to make sure the server is available (green icon). It might take some time for the server to start (for example, 15 minutes). Then click **Save**.

Switch from webMethods Broker to Universal Messaging

If you use webMethods Broker as your JMS provider in the old release, and are going to use Universal Messaging in the new release, do the following:

1. Start Universal Messaging and open Universal Messaging Enterprise Manager.

2. Connect to a Universal Messaging instance, then click it.

3. Click the **Config** tab. Expand **Global Values** and set AllowRealmAdminFullAccess to true.

14 Migrate the Asset Build Environment, Deployer, and Application Platform

Migrate the Asset Build Environment

Go to the *old_Software AG_directory*/common/AssetBuildEnvironment/master_build directory and copy the build.properties file to the same location in the new installation.

Migrate Deployer

Configurations, Data, and Assets that Will be Migrated

- Settings.
- Server aliases.
- Target groups.
- Projects.

Note:
Migrating Deployer projects does not convert the product assets in those projects from their old release to the new release. You must migrate the assets as instructed earlier in this guide.

Migrate Deployer

1. Upgrade all source and target product servers that were defined in the old Deployer to the new release, and migrate all configurations, data, and assets for those products to the new release, as instructed earlier in this guide.

2. If the new and old Deployer are on the same machine, make sure the old Deployer is shut down.

3. If Deployer is hosted on Microservices Runtime, migrate Deployer and the host Microservices Runtime using the Integration Server migration utility (see "Migrate Products Using Migration Utilities" on page 121).

   If Deployer is hosted on Integration Server, you can migrate Deployer and the host Integration Server using the Integration Server migration utility (see "Migrate Products Using Migration Utilities" on page 121) or Command Central (see "Migrate Integration Server, Microservices Runtime, and Hosted Wm Packages" on page 67).

   Note:
   Do not start the new host before you migrate Deployer. If you do, a file named MIGRATION_DONE will be created in the WmDeployer package, and you will not be able to perform the migration. If you did start the new host, delete this file.

Migrate Application Platform

Application Platform Configurations, Data, and Assets that Will be Migrated

- Application Platform configuration files that reside in Integration Server installations.
- Application Platform projects bundles that reside in My webMethods Server installations.
- Upgrade from 9.9, 9.10, or 9.12: Web applications that were created using WmTomcat for use as Application Platform web applications. The web applications will be migrated as WAR files to the *new_Software AG_directory*/profiles/*instance_name*/workspace/webapps directory.
- Upgrade from 10.1 or 10.3: Web applications that were deployed in the webapps directory of Integration Server instances that host the Application Platform Support package.

Migrate Application Platform

1. If the new and old Integration Servers that host the Application Platform Support package are on the same machine, make sure the old Integration Servers are shut down.

2. If the new and old My webMethods Servers that host Application Platform projects bundles are on the same machine, make sure the old My webMethods Servers are shut down.

3. You can migrate using either of two methods.

   - To migrate using the migration utility, follow the instructions in "Migrate Products Using Migration Utilities" on page 121.
   - To migrate using Command Central, use the Command Central command below.

     ```
     sagcc exec administration product target_node_alias
     PLSCore migration migrate
     {srcDir|srcFile}=full_path_to_{old_Software AG_directory|ZIP file}
     ```

Complete the Application Platform Upgrade

Verify and Correct Package Imports

Go to the *new_Software AG_directory*/install/log directory and open the migrationLog.txt file. Check for messages similar to this message:

```
Project bundle /dev/installs/99oct2015/profiles/IS_default/
workspace/app-platform/deployer/bundles/greeter-web.jar
contains a package com.softwareag.applatform.sdk with import version
range [start_version, end_version) that must be upgraded manually.
```

The message relates to semantic versioning of the packages used in your Application Platform projects. For information, see the OSGi Alliance semantic versioning technical whitepaper.

If the message exists, and you never created manifest files for Application Platform projects, rebuild and redeploy the projects using Asset Build Environment and Deployer.

If the message exists, and you did create manifest files for Application Platform projects, edit the manifest files to include the *end_version*, and then rebuild and redeploy the projects.

For instructions on rebuilding and redeploying Application Platform projects , see *webMethods Application Platform User's Guide*.

Update the Tomcat Configuration

1. Go to *old_Software AG_directory*/IntegrationServer/web/conf directory and open all configuration files in that directory in a text editor.

2. Go to *new_Software AG_directory*/profiles/*instance_name*/configuration/tomcat/conf directory and open all configuration files in that directory in a text editor.

3. Go to https://tomcat.apache.org/migration-85.html.

4. Using the instructions on the web page, modify the new configuration files to have the same custom changes you made in the old configuration files.

5. Go to the *new_Software AG_directory*/profiles/*instance_name*/configuration/tomcat/conf directory, open the server.xml file, and replace the existing <Realm> element with the following:

```
<Realm className="org.apache.catalina.realm.LockOutRealm">
  <Realm className="com.softwareag.platform.catalina.auth.SINRealm"
     name="AppPlatformRealm"
     userClassNames="com.softwareag.security.jaas.principals.SagUserPrincipal"

     roleClassNames="com.softwareag.security.jaas.principals.SagRolePrincipal"

     defaultRealm="AppPlatformRealm"/>
</Realm>
```

6. Review your web applications to make sure they comply with Tomcat server 8.5 and servlet API 3.1.

Upgrade from 9.9, 9.10, or 9.12: Complete the WmTomcat Web Applications Migration

1. Remove the dependency of each web application on WmTomcat. For each migrated Integration Server package, go to the *new_Software AG_directory*/IntegrationServer/instances/*instance_name*/*migrated_package* directory, open the manifest.v3 file in a text editor, remove the line below, and then save the file.

```
<value name="WmTomcat">...</value>
```

2. The URL for the migrated web applications is different from the URL for the original web applications; the port number changes, and the /web segment is removed (for example, the old URL http://localhost:5555/web/MyPackage/index.jsp becomes http://localhost:8072/MyPackage/index.jsp). Notify your users and update your software components that depend upon the URL.

3. Each web application was migrated as a war ZIP file. If you want to reduce the time it takes the Tomcat servlet container to start, go to the *new_Software AG_directory*/profiles/*instance_name*/workspace/webapps directory and extract the war files from each ZIP file into the webapps directory.

4. If you have direct calls to the Integration Server API com.wm.app.b2b.server.Service.doInvoke( ) in the JSP page for a web application, replace them with `<webm:invoke/>` custom tags. For instructions, see the Application Platform documentation.

15 Migrate Apama

1. If the new and old Apama are on different machines, update deployment scripts (for example, Ant scripts and properties files) to specify the correct host names for the new installation.
2. If the new and old Apama are on the same machine, make sure all processes from the old installation are shut down.
3. Re-run any build or deployment scripts to make sure generated artifacts such as Apama queries, plug-ins, and scenarios from your projects are up to date. This task could include re-exporting and re-running Ant scripts, executing engine_deploy, rebuilding Docker images, and re-running custom scripts. Review the output for errors and warnings.
4. Inject your migrated applications into the new Apama by running them in your user acceptance testing (UAT) environment. Scan the correlator logs for any errors or warnings. Re-run all your system tests (for example, PySys test cases) and ensure your application behaves as expected.
5. If you are using Command Central, re-create Apama instances in the new installation using the same command line, GUI, or composite template procedure you used to create the instances in the old installation, with modifications to accommodate any deprecated or removed items mentioned in the Apama release notes.

16 Migrate API Gateway, API Portal, and CentraSite

Migrate API Gateway

API Gateway Configurations, Data, and Assets that Will be Migrated

- Data store. The data store contains the following:
  - APIs.
  - Applications.
  - Policies and policy actions.
  - Configurations and administrator settings.
  - Aliases.
  - Users, groups, and access profiles.
  - Plans, packages, and subscriptions.
  - API events for transactions, monitoring, policy violations, threat protection, lifecycle, and errors.
  - Performance metrics.
- Configurations for the old Integration Server that hosts API Gateway, the API Gateway UI and dashboard (Kibana), and for one API Gateway instance.

Migrate API Gateway

For instructions on upgrading API Gateway, see API Gateway Migration steps.

Migrate API Portal

API Portal Assets that Can be Migrated

- Users and user groups.
- APIs and API documents.
- Plans and packages.
- Apps and applications.
- Collaboration data.
- Upgrade from 10.1: Analytics data.

Migrate API Portal

1. If the new and old API Portals are on the same machine, make sure the old API Portal is shut down.

2. If you installed the new API Portal 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 cloud agent port is 18012.

   The table below provides details about the REST service.

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": "Clous", "password": "g3h31m“}`

3. Start the new API Portal Cloud Controller.

4. Run this command:

```
acc> startall
```

5. For each tenant in the old API Portal, create a corresponding tenant in the new API Portal and then import the license for the new tenant. For instructions, see *webMethods API Portal Administrator's Guide*.

6. For each new tenant you created, restore the data for the corresponding old tenant into the new API Portal 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. Only the user metadata will be merged.

   The table below explains how to restore each old tenant's data into the new API Portal for each old product release.

For	Run these commands...

9.9
9.10
9.12
	`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`
	`acc> invoke prepareTenant on bundle_runnable_name tenant.name=tenant_name`

10.1 10.3 10.4 10.5	`acc> restore tenant tenant_name from full_path_to_backup_file username=your_user_name password=your_password` `acc> invoke prepareTenant on bundle_runnable_name tenant.name=tenant_name`

Complete the API Portal Upgrade

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*.

Migrate CentraSite

CentraSite Configurations that Will be Migrated

- Configurations from the old CentraSite and assets are migrated from the old Registry Repository to the new CentraSite installation.
- LDAP configuration is transformed and migrated from the old Registry Repository to the new CentraSite JAAS configuration.

Migrate CentraSite

Import configurations and assets from the ZIP file you created earlier. Open a command window or shell, go to the *new_Software AG_directory*/CentraSite/utilities directory, and run this command:

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

For example:

```
./sbsImport.sh /tmp/sbs_cs82_data.zip
```

Complete the CentraSite Upgrade

1. Start the new CentraSite.

2. 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*/profiles/CTP/configuration directories. Copy the following from the old file to the new 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 new 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 CentraSite 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.

3. Change any settings in CentraSite Control that you noted earlier.

4. If you installed plug-ins that are GUI extensions for CentraSite Control in the old CentraSite installation, install them in the new CentraSite installation.

5. If you are using API Portal with CentraSite or API Gateway, republish all API Portal instances you created in the old release to the new API Portal. For instructions, see the CentraSite or API Gateway documentation.

17 Migrate MashZone NextGen

Note:
There is no direct path fromMashZone NextGen 9.9 to MashZone NextGen 10.3. Instead, you must upgrade to MashZone NextGen 9.10 using the instructions in the 9.10 upgrade guide, and then upgrade from 9.10 to MashZone NextGen 10.3 using the instructions in this guide.

Note:
MashZone NextGen was renamed MashZone NextGen Business Analytics in release 9.12, then renamed MashZone NextGen in release 10.1.

Upgrade from 10.1 or 10.3: Migrate MashZone NextGen

MashZone NextGen Configurations, Data, and Assets that Will be Migrated

- Database you were using for the old repository. If you used the embedded Derby database, the utility migrates the database. If you used an external database, the utility migrates the database driver, the MashZone NextGen repository libraries, and the database.
- LDAP configuration.
- User-defined custom look and feel, user-defined custom widgets, and RAQL user-defined functions (UDFs) that are stored in the *old_Software AG_directory*/MashZoneNG/raql-udfs directory.
- Data files you defined and stored in the *old_Software AG_directory*/MashZoneNG/mashzone/data/resources directory.
- Ehcache attribute information that you changed in the old installation.
- Properties required to keep the new installation running properly. Port values are not migrated.

Migrate MashZone NextGen

1. If you stored data files in a directory other than the default (that is, *old_Software AG_directory*/MashZoneNG/mashzone/data/resources) and you want to migrate those data files, move them into the default directory.

2. If you stored RAQL UDFs in a directory other than the default (that is, *old_Software AG_directory*/MashZoneNG/raql-udfs) and you want to migrate those UDFs, move them into the default directory.

3. If the new and old MashZone NextGens are on the same machine, make sure the old MashZone NextGen is shut down.

4. You can migrate using either of two methods.

   - To migrate using the migration utility, follow the instructions in "Migrate Products Using Migration Utilities" on page 121.

- To migrate using Command Central, use the Command Central command below.

```
sagcc exec administration product target_node_alias
Presto migration migrate
{srcDir|srcFile}=full_path_to_{old_Software AG_directory|ZIP file}
```

Complete the MashZone NextGen Upgrade

Before you start MashZone NextGen, do the following if necessary:

- If you changed the MashZone NextGen startup files in the old installation and you want to re-use them in the new installation, go to the *old_Software AG_directory*/MashZoneNG/apache-tomcat/bin and copy the files to the same location in the new installation. If the files specify absolute paths, update the paths to point to the correction locations in the new installation.
- The old presto.config file references custom files such as truststores, keystores, and the landingpage welcometext file. If you changed these references or the referenced files in the old installation, go to the *new_Software AG_directory*/MashZoneNG/apache-tomcat/webapps/mashzone/WEB-INF/classes directory, open the presto.config file, and check whether the referenced files are available in the new installation. If they are not, make them available.

Upgrade from 9.10 or 9.12: Migrate MashZone NextGen

Set up the new MashZone NextGen to use the old repository. If you deployed extensions to features of the old product, you will also copy the extensions to the new MashZone NextGen installation.

Set Up the MashZone NextGen Repository

Set Up Repository When Using the Embedded Derby Database

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

1. The new MashZone NextGen comes with a pre-populated Derby repository. Go to the *new_Software AG_directory*/MashZoneNG/apache-tomcat/bin and rename the mashzonenextgenrepository directory (for example, rename the directory mashzonenextgenrepository_orig).

2. Upgrade from 9.10 or 9.12: Go to the *old_Software AG_directory*/MashZoneNG/apache-tomee-jaxrs/bin directory and copy the mashzonenextgenrepository and mashzonerepository directories to the *new_Software AG_directory* /MashZoneNG/apache-tomcat/bin directory.

3. Execute Derby database update SQL commands for the new release. You can execute these commands using any SQL client that can connect to Derby, or you can use the Derby IJ utility delivered with MashZone NextGen, as follows:

a. Open a command window and change to the directory where the Derby database resides (for example, cd *new_Software AG_directory*/MashZoneNG/apache-tomcat/bin).

b. Start the IJ utility by running this command:

```
java -jar SAG_HOME/MashZoneNG/prestorepository/derby/lib/derbyrun.jar ij
```

c. Connect to the embedded Derby database by running this command:

```
connect 'jdbc:derby:mashzonenextgenrepository';
```

d. Run the file using the RUN command. For example:

```
RUN 'new_Software
AG_directory/MashZoneNG/upgrade/10.1/MIGRATE-quartz2-DERBY.sql';
```

e. Exit the IJ utility by running the command `EXIT;`

Set Up Repository When Using Any Other Database

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

1. Go to the *old_Software AG_directory*/{Presto|MashZoneNG}/apache-tomee-jaxrs/lib directory and copy the JDBC driver jar file for your RDBMS to the *new_Software AG_directory*/MashZoneNG/apache-tomcat/lib directory.

   The table below lists the JDBC driver jar file to copy for each type of database.

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 the old product, do the following:

   a. Go to the *new_Software AG_directory*/{Presto|MashZoneNG}/apache-tomcat/webapps/mashzone/WEB-INF/lib directory and delete the jackbe-presto-rds-postgre-derby-10.3.jar file.

   b. Go to the *new_Software AG_directory*/{Presto|MashZoneNG}/prestorepository directory and copy the jackbe-presto-rds-oracle-mysql-mssql-10.3.jar file to the *new_Software AG_directory*/MashZoneNG/apache-tomcat/webapps/mashzone/WEB-INF/lib directory.

3. Configure the new MashZone NextGen to use the existing repository. For instructions, see *MashZone NextGen Administration Guide*, and then see the section on getting started with MashZone NextGen, and then see the section on moving the MashZone NextGen repositories to a robust database solution. Skip steps 1 through 4, and start with step 5.

   Note:
   In MashZone NextGen 10.0 and earlier, repository connection information was defined in the apache-tomee-jaxrs/conf/tomee.xml file. In the new MashZone NextGen release, repository connection information was moved to the apache-tomcat/conf/context.xml file and the format has changed.

4. Execute database update SQL commands for the new release. Use a SQL client that is connected to your mashzonenextgenrepository. If you need help, contact Software AG Global Support.

   a. Go to the *new_SoftwareAG_directory*/MashZoneNG/upgrade/10.1 directory and open the appropriate *database_type*.xml file in a text editor.

   b. Read the instructions in the file and modify SQL statements that remove existing constraints as necessary.

   c. Execute the SQL statements in the file one by one.

Upgrade from 9.10: Add Data from Old Repository to New Repository

MashZone NextGen 9.10 had two repositories, mashzonerepository and mashzonenextgenrepository. Starting with MashZone NextGen 9.12, there is only one repository, mashzonenextgenrepository. The new mashzonenextgenrepository already contains the data from the old mashzonenextgenrepository, but you must manually add the data from the old mashzonerepository.

1. Go to the *new_Software AG_directory*/MashZoneNG/apache-tomcat/conf directory and open the context.xml file. Add a <Resource> entry that sets the connection attributes properly for the old mashzonerepository. For example:

```
<Resource name="amzDatabase" auth="Container" type="javax.sql.DataSource"
  maxTotal="200"
  maxIdle="30"
  maxWaitMillis="10000"
  validationQuery="values(1)"
  username="app"
  password="app"
driverClassName="org.apache.derby.jdbc.EmbeddedDriver"
  url="jdbc:derby:
${catalina.base}/bin/mashzonerepository"/>
```

2. Go to the *new_Software AG_directory*/MashZoneNG/apache-tomcat/webapps/mashzone/WEB-INF directory, open

the web.xml file in a text editor, and add this XML `<resource-ref>` element to the bottom of the file, just before the closing `</web-app>` tag:

```
<resource-ref>
  <description>DB Connection</description>
  <res-ref-name>amzDatabase</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

3. Start the MashZone NextGen server.

   Data from the amzDatabase will automatically be migrated into the mashzonenextgenrepository, and amzDatabase will no longer be used.

4. Shut down the server.

5. Go to the *new_Software AG_directory*/MashZoneNG/apache-tomcat/webapps/mashzone/WEB-INF directory and open the web.xml file in a text editor. Remove the `<resource-ref>` element you just added and save the file.

6. Go to the *new_Software AG_directory*/MashZoneNG/apache-tomcat/conf directory and open the context.xml file. Remove the `<Resource id="amzDatabase">` element and save the file.

Update the MashZone NextGen Configuration and Copy Extensions

Most of the MashZone NextGen configuration is stored in the MashZone NextGen 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 product. The procedure varies depending on whether you used an external configuration directory with the old product.

Update When an External Configuration Directory Was Used

1. Update the classpath in the application server that hosts the new MashZone NextGen to work with the external configuration directory and any subdirectories. For instructions, see the topic on setting up an external configuration directory in the *MashZone NextGen Administration Guide*.

2. Manually migrate the changes you made in the old configuration files listed below from the old product to the corresponding files in the new MashZone NextGen. 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 *MashZone NextGen Administration Guide*.

- 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 product to the corresponding directory for the new MashZone NextGen. See the *MashZone NextGen Administration Guide* 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 product to the corresponding files in the new MashZone NextGen. 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 *MashZone NextGen Administration Guide*.
   - The userRepositoryLdap.properties file, located in the *old_Software AG_directory*/{Presto|MashZoneNG}/apache-tomee-jaxrs/webapps/mashzone/WEB-INF/classes directory.

Upgrade from 9.10: Migrate Feeds and Run Upgrade Commands

1. Data feeds that were created with the 9.10 MashZone Feed Editor are not supported in MashZone NextGen 9.12 and later. If you want to use these data feeds you must export them from the old installation and then import them into the new installation. For instructions on exporting and importing data feeds, see the *MashZone NextGen Administration Guide*.

2. If the MashZone NextGen server is not running, start it.

3. Open a command window or shell, go to the *new_Software AG_directory* /MashZoneNG/prestocli/bin directory, and run this command:

```
./padmin.bat runRequestFile -u Administrator -c -w manage
-f ../../upgrade/9.12/upgrade_9.12.0.jump.txt
```

18 Upgrade from 9.9: Migrate Mobile Designer

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 delete the MOBILE_DESIGNER environment variable, do the following:

- For 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.
- For 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.

19 Migrate OneData

OneData Configurations, Data, and Assets that Will be Migrated

- Configuration data except for Log4j properties.

Migrate OneData

1. If the new and old OneDatas are on the same machine, make sure the old OneData is shut down.

2. Run the OneData migration utility as described in "Migrate Products Using Migration Utilities" on page 121.

Complete the OneData Upgrade

Perform the tasks below as appropriate.

- If you customized the old Java Service Wrapper, go to the *Software AG_directory*/profiles/ODE/configuration directories in the old and new installations, open the custom_wrapper.conf files, and copy the customizations from the old file to the new file.

  Note:
  As noted in the *Software AG Infrastructure Administrator's Guide*, you should never modify the wrapper.conf file unless instructed to do so by Software AG. If you mistakenly modified the old wrapper.conf file, manually copy values for properties you modified in the old wrapper.conf file to the corresponding properties in the new custom_wrapper.conf file.

- If you customized the old Log4j, go to the *Software AG_directory*/profiles/ODE/bin/onedata/config directory in the old and new installations, open the log4j.properties files, and copy the customizations from the old file to the new file.

- Go to the *Software AG_directory*/profiles/ODE/bin/onedata/config directories in the old and new installations and open the onedata.properties file. If the old onedata.search.elasticsearch.indexanalyzer property was set to autocomplete_analyzer, set the new onedata.search.elasticsearch.indexanalyzer property to autocomplete.

- If you are using JMS with OneData, copy the client jar files you backed up earlier to the *new_Software AG_directory*/profiles/ODE/webapp/onedata/WEB-INF/lib directory.

- 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*.

Adjust the OneData Metadata schema.

1. Log in to the Metadata schema.

2. Check the number of columns in the OD_MD_QUEUE_RCD table. If your table has COLUMN_1 through COLUMN_200, skip this step.

- If the table has the default of 50 columns, extend the table by running `ALTER TABLE OD_MD_QUEUE_RCD ADD`
- If the table has more than 50 columns but less than 200, add only the columns required to reach 200.
  - For Oracle, each column should look like `COLUMN_<number> varchar2(2000 byte,` and the last line should be `;`
  - For SQL Server, each column should look like `COLUMN_<number> [nvarchar](2000) NULL,` and the last line should be `;`

3. Go to the *Software AG_directory*/profiles/ODE/bin/onedata/config directory in the new installation and open the onedata.properties file. Set the two properties shown below as follows:.

   onedata.exceptionqueue.fetchsize=50

   onedata.exceptionqueue.numberofcolumns=200

20 Migrate Zementis Predictive Analytics

Zementis Predictive Analytics Assets that Will be Migrated

The Zementis Predictive Analytics migration script described below will migrate Zementis Predictive Analytics models and resources.

Migrate Zementis Predictive Analytics

1. If the new and old Zementis Predictive Analytics servers are on the same machine, make sure the old Zementis Predictive Analytics server is shut down.

2. If the old and new installations are on different machines, copy the adapa-backup-*yyyymmdd-hhmmss* directory from the old installation to the machine that hosts the new installation.

3. Go to the *new_Software AG_directory*/Zementis/adapa-app/migration directory.

4. Restore Zementis Predictive Analytics models and resources by running this command:

```
./migration.sh restore Zementis_host_URL admin_user admin_password
full_path_to_adapa-backup-yyyymmdd-hhmmss_directory
```

21 Migrate Digital Event Services

Migrate Digital Event Services

Digital Event Services components are automatically installed with products, except for Microservices Runtime. These components are the Digital Event Services runtime and the Digital Event Services Type Repository.

Digital Event Services Configurations, Data, and Assets that Will be Migrated

- Digital Event Services runtime configurations for all products.
- Digital Event Services Type Repository.
- If you used Digital Event Persistence to persist events, the service configurations.

Migrate Digital Event Services

1. Make sure none of the new products are running.
2. You can migrate using either of two methods.
   - To migrate using the migration utility, follow the instructions in "Migrate Products Using Migration Utilities" on page 121.
   - To migrate using Command Central, use the Command Central command below.

```
sagcc exec administration product target_node_alias
DEVRuntime migration migrate
{srcDir|srcFile}=full_path_to_{old_Software AG_directory|ZIP file}
```

Complete the Digital Event Services Upgrade

The Digital Event Services truststore and keystore configurations might contain references to external resources. If those resources are in different locations in relation to the new installation, go to the *new_Software AG_directory*/profiles/product/configuration/DigitalEventServices/truststores and keystores directories, respectively, open the configuration files in a text editor and update the references. The references are specified on the location field.

22 Migrate Products Using Migration Utilities

Before You Run Migration Utilities

Most products offer migration utilities that automatically migrate configurations and 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.

You can run migration utilities as follows:

- For some products, migration utilities run without prompting you for any information (that is, silently). If the utility fails to migrate an item, the utility exits. The utility does not revert the new product installation. You can address the issue and rerun the utility.
- For other products, you can run migration utilities as follows:
  - You can run a custom migration, in which you select the configurations and data to migrate. The utility gathers your settings through a series of prompts, then migrates the selected configurations and data. You can export your settings to a file named migrate.dat and use them in other upgrades.
  - You can run a migration with imported settings and some prompting or without any prompting (that is, silently). The imported settings can come from settings you exported from a custom migration, or from the default migration provided by Software AG with the product installation in a file named migrate*old_release*sbs.dat. The settings for default migrations are described in the product-specific chapters.

  If you are running the utility with prompting, and the utility fails to migrate an item, the utility asks whether to continue with the next item or abort the migration. If you choose to abort, the utility exits. The utility does not revert the new product installation. You can address the issue and rerun the utility.

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

Migration utilities write detailed migration information to the command window and to the migrationLog.txt file in the *new_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 log4j2 file (for example, the *new_Software AG_directory*/*product*/bin/migrate, or /migrate/bin, or bin/migrate/resources directory), open the file in a text editor, set the `logger.1.level` property to `DEBUG` and save and close the file.

Run the Software AG Infrastructure Migration Utility

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

1. On the machine that hosts the new products, open a command window or shell, go to the *new_Software AG_directory*/common/migrate/osgi/bin directory, and run the command below. For the migrate*old_release*sbs.dat file, specify *old_release* without periods (for example, 980, or 9100, or 1030).

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

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

3. Install the latest product fix on the new Common Platform and the new Security Infrastructure. Fix names for these products typically include the letters OSGI and SIN, respectively. For instructions on installing fixes, see *Using Software AG Update Manager* and the fix readme files.

Run the Universal Messaging Migration Utility

Perform a Custom Migration

1. On the machine that hosts the new Universal Messaging, open a command window or shell, go to the *new_Software AG_directory*/UniversalMessaging/tools/migrate directory, and run the command `migrate.{bat|sh}`.

2. The utility asks for the full path to the Software AG directory that contains the old Universal Messaging installation or the ZIP file you made earlier.

3. The utility asks whether to import migration settings. Enter N.

4. The utility asks which Universal Messaging instance to migrate and lists the instances in the old installation. To specify multiple instances, separate them using commas.

   To migrate all instances, press Enter without specifying any instances.

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 can come from the following:

- Settings you exported from a custom migration. These settings are stored in a file named migrate.dat in the *new_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 new Universal Messaging installations to which you want to migrate data.
- Settings in the default migrations provided with Universal Messaging. For each old release, the settings are stored in a file named migrate*old_release*sbs.dat file provided by Software AG in the *new_Software AG_directory*/UniversalMessaging/tools/migrate directory. The settings tell the migration utility to migrate the data listed under " Universal Messaging Configurations, Data, and Assets that Can be Migrated" on page 60.

Migrate Using Custom Imported Settings with Prompting

1. On the machine that hosts the new Universal Messaging, open a command window or shell, go to the *new_Software AG_directory*/UniversalMessaging/tools/migrate directory, and run the command `migrate.{bat|sh}`.

2. The utility asks for the full path to the Software AG directory that contains the old Universal Messaging installation or to the ZIP file you made earlier.

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 Custom Imported Settings without Prompting (Silent)

On the machine that hosts the new Universal Messaging, open a command window or shell, go to the *new_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_Software AG_directory|ZIP file}
-importFile full_path_to_migrate.dat
-silent true
```

Migrate Using Default Imported Settings without Prompting (Silent)

On the machine that hosts the new Universal Messaging, open a command window or shell, go to the *new_Software AG_directory*/UniversalMessaging/tools/migrate directory, and run the command below. For the migrate*old_release*sbs.dat file, specify *old_release* without periods (for example,980, or 9100, or 1030).

The default imported settings migrate all instances, therefore only specify `-instanceName` if you want to override that default.

If an error occurs, the utility exits.

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Software AG_directory|ZIP file}
-importFile migrateold_releasesbs.dat
[-instanceName old_instance_name[,old_instance_name...]]
-silent true
```

Run the My webMethods Server Migration Utility

Perform a Custom Migration

1. On the machine that hosts the new My webMethods Server, open a command window or shell, go to the *new_Software AG_directory*/MWS/bin/migrate directory, and run the command `migrate.{bat|sh}`.

2. The utility asks for the full path to the Software AG directory that contains the old My webMethods Server installation or the ZIP file you made earlier.

3. The utility asks whether to import migration settings. Enter N.

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

   a. Whether to migrate the instance.

   b. Whether to use the live database or a cloned database with the migrated instance. If you respond that you are using a cloned database, the utility prompts for the database URL, user, and password.

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

   c. If the old instance was connected to Universal Messaging, whether to use the old Universal Messaging URL or a new URL for the migrated instance. If you respond that you want to use a new URL, the utility prompts for the URL.

      If the old instance was not connected to Universal Messaging, whether to specify a connection to Universal Messaging now or at some later time. If you respond that you want to specify the connection now, the utility prompts for the Universal Messaging URL.

   d. Whether to use a new node name for the migrated instance.

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 instances you specified.

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 *new_Software AG_directory*/MWS/bin/migrate directory. Copy the migrate.dat

file to any directory on machines that host new My webMethods Server installations to which you want to migrate data.

- Settings in the default migrations provided by Software AG with My webMethods Server. For each old release, the settings are stored in a file named migrate*old_release*sbs.dat file in the *new_Software AG_directory*/MWS/bin/migrate directory. The settings tell the migration utility to migrate all instances within the old installation to the new installation, and to use the live database.

Migrate Using Custom Imported Settings with Prompting

1. On the machine that hosts the new My webMethods Server, open a command window or shell, go to the *new_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 earlier.

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 Default Imported Settings without Prompting (Silent)

You can migrate in silent mode when the old and new My webMethods Server installations are identical (for example, they are hosting the same user interfaces for other products).

On the machine that hosts the new My webMethods Server, open a command window or shell, go to the *new_Software AG_directory*/MWS/bin/migrate directory, and run the command below. For the migrate*old_release*sbs.dat file, specify *old_release* without periods (for example, 980, or 9100, or 1030).

The default imported settings migrate all instances, therefore only specify `-instanceName` if you want to override that default and migrate specific instances only.

If an error occurs (for example, the installations are not identical), the utility exits.

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Software AG_directory|ZIP file}
-importFile migrateold_releasesbs.dat
[-cloneDbURL URL -cloneDbUser user -cloneDbPassword password]
[-instanceName old_instance_name[,old_instance_name...]]
[-newNodeName old_instance_name:new_node_name[,old_instance_name:new_node_name,...]]
-silent true
```

Migrate Using Custom Imported Settings without Prompting (Silent)

You can migrate in silent mode when the old and new My webMethods Server installations are identical (for example, they are hosting the same user interfaces for other products).

On the machine that hosts the new My webMethods Server, open a command window or shell, go to the *new_Software AG_directory*/MWS/bin/migrate directory, and run the command below.

If an error occurs (for example, the installations are not identical), 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
```

Run the Integration Server Migration Utility to Migrate Integration Server or Microservices Runtime

If the old installation had multiple server instances, run the migration utility once for each instance to migrate.

If you are migrating Integration Server, you can migrate an old instance to the new instance created during installation, to a new instance you created after installation, or to a new instance the utility creates for you.

If you are migrating Microservices Runtime, migrate each instance to a separate Microservices Runtime installation. Starting in 10.4, each Microservices Runtime supports only one server instance.

Perform a Custom Migration for Integration Server

1. On the machine that hosts the new installation, open a command window or shell and go to the *new_Software AG_directory*/IntegrationServer/bin/migrate directory.

2. Run the command `migrate.{bat|sh}`.

3. The utility asks for the full path to the Software AG directory that contains the old installation or the ZIP file you made earlier.

4. If the old installation has more than one server instance, the utility asks which instance to migrate and lists the instances in the old installation.

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

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

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 the password store, and whether to migrate passwords.

8. The utility asks whether to migrate the 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.

9. The utility asks you to specify the behavior to use for new properties that have been added to the new installation and existing properties that have new defaults in the new installation. If you choose to specify the behavior to use, the utility lists each property and asks whether to use the new behavior or preserve existing behavior. You can read about these properties in the Integration Server readme.

   For CloudStreams, select the new behavior `asNeeded` for the watt.server.http.listRequestVars property.

10. The utility asks whether to migrate functional aliases, database driver configurations, and JDBC connection pool configurations. If you enter Y for JDBC connection pool configurations, the utility asks whether to use the live database or a cloned database with each migrated connection pool. If you respond that you are using a cloned database for the migrated connection pool, the utility asks for the database URL, user, and password.

11. If the Integration Server hosts Wm packages, the utility asks whether to migrate the configuration files for those packages. For some products, you might see additional prompts for migrating other data.

12. 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.

13. 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 the new Trading Networks, and replaces the properties.cnf file in the *new_Software AG_directory*/IntegrationServer/instances/*instance_name*/packages/WmTN/config directory.

    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.

14. The utility asks whether to migrate JAAS configurations.

15. The utility asks whether to migrate Java Service Wrapper customizations you made in the old custom_wrapper.conf files, if any. These are migrated to the new custom_wrapper.conf file.

16. The utility asks whether to update the host name in your database tables. If the old and new installations are on different machines, respond Y.

17. 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.

Note:
When cloned database configurations are provided, out of security concerns the cloned database password is not exported into the settings. If you do not continue the migration, you must specify `cloneDbPassword` in the command line when you use the custom settings to perform a migration.

18. The utility asks whether to begin migration. If you enter Y, the utility migrates the data you selected.

Perform a Custom Migration for Microservices Runtime

1. On the machine that hosts the new installation, open a command window or shell and go to the *new_Software AG_directory*/IntegrationServer/bin/migrate directory.

2. Run the command `migrate.{bat|sh}`.

3. The utility asks for the full path to the Software AG directory that contains the old installation or the ZIP file you made earlier.

4. If the old installation has more than one server instance, the utility asks which instance to migrate and lists the instances in the old 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 selected packages only, the utility lists each package and asks whether to migrate it.

7. The utility asks whether to migrate the password store, and whether to migrate passwords.

8. The utility asks whether to migrate the Microservices Runtime 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.

9. The utility asks you to specify the behavior to use for new properties that have been added to the new installation and existing properties that have new defaults in the new installation. If you choose to specify the behavior to use, the utility lists each property and asks whether to use the new behavior or preserve existing behavior. You can read about these properties in the Microservices Runtime readme.

   For CloudStreams, select the new behavior `asNeeded` for the watt.server.http.listRequestVars property.

10. The utility asks whether to migrate functional aliases, database driver configurations, and JDBC connection pool configurations. If you enter Y for JDBC connection pool configurations, the utility asks whether to use the live database or a cloned database with each migrated connection pool. If you respond that you are using a cloned database for the migrated connection pool, the utility asks for the database URL, user, and password.

11. If the Microservices Runtime hosts Wm packages, the utility asks whether to migrate the configuration files for those packages. For some products, you might see additional prompts for migrating other data.

12. 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.

13. The utility asks whether to update the host name in your database tables. If the old and new installations are on different machines, respond Y.

14. 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.

    Note:
    When cloned database configurations are provided, out of security concerns the cloned database password is not exported into the settings. If you do not continue the migration, you must specify `cloneDbPassword` in the command line when you use the custom settings to perform a migration.

15. 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 in the *new_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 the new Integration Server installations to which you want to migrate data.
- Settings in the default migrations provided with Integration Server or Microservices Runtime. For each old release, the settings are stored in a file named migrate*old_release*sbs.dat file provided by Software AG in the *new_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 to Migrate Integration Server or Microservices Runtime" on page 127. For properties that are new or that have new defaults, the default migration chooses the behavior that best preserves backwards compatibility.

  Note:
  When you use custom migration settings with a cloned database, you must specify `cloneDbPassword` in the command line to perform the migration.

Migrate Integration Server Using Custom or Default Imported Settings with Prompting

1. On the machine that hosts the new installation, open a command window or shell, go to the *`new_Software AG_directory`*/IntegrationServer/bin/migrate directory, and run the command `migrate.{bat|sh}`.

2. The utility asks for the full path to the Software AG directory that contains the old installation or to the ZIP file you made earlier.

3. If the old installation has more than one server instance, the utility asks which instance to migrate and lists the instances in the old installation.

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

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

4. 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*`old_release`*sbs.dat file. For the migrate*`old_release`*sbs.dat file, specify *`old_release`* without periods (for example, 980, or 9100, or 1030).

Migrate Microservices Runtime Using Custom or Default Imported Settings with Prompting

1. On the machine that hosts the new installation, open a command window or shell, go to the *new_Software AG_directory*/IntegrationServer/bin/migrate directory, and run the command `migrate.{bat|sh}`.

2. The utility asks for the full path to the Software AG directory that contains the old installation or to the ZIP file you made earlier.

3. If the old installation has more than one server instance, the utility asks which instance to migrate and lists the instances in the old installation.

4. 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*old_release*sbs.dat file. For the migrate*old_release*sbs.dat file, specify *old_release* without periods (for example, 980, or 9100, or 1030).

Migrate Using Custom or Default Imported Settings without Prompting (Silent)

If you are using a cloned database for Integration Server or Microservices Runtime database components, specify the -cloneDb options on the command as shown below.

Note:
The value you specify for `cloneDbPassword` is not stored in the migrate.bat file.

The migration utility uses the -cloneDb options only for the Integration Server or Microservices Runtime database components. If you are using a cloned database for the database components of hosted products (for example, Trading Networks), perform a custom migration that identifies the cloned database, then run the command below with the migrate.dat file created by the custom migration.

On the machine that hosts the new installation, open a command window or shell, go to the *new_Software AG_directory*/IntegrationServer/bin/migrate directory, and run the command below. For the migrate*old_release*sbs.dat file, specify the source release number without periods (for example, 980, or 9100, or 1030).

If you want to specify a new name for the migrated instance, specify -newInstanceName.

If an error occurs, the utility exits.

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Integration Server_or_
Microservices_Runtime_directory|ZIP_file}
-importFile {migrateold_releasesbs.dat|full_path_to_migrate.dat}
-instanceName old_instance_name [-newInstanceName new_instance_name]
[-cloneDbURL URL -cloneDbUser user -cloneDbPassword password]
-silent true
```

Run the Infrastructure Data Collector Migration Utility

Perform a Custom Migration

1. On the machine that hosts the new Infrastructure Data Collector, open a command window or shell, go to the *new_Software AG_directory*/Infrastructuredc/bin/migrate directory and run the command `migrate.{bat|sh}`.

2. The utility asks for the full path to the Software AG directory that contains the old Infrastructure Data Collector installation or the ZIP file you made earlier.

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 new Infrastructure Data Collector installation, the utility backs up the new file to the *new_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 *new_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 new Infrastructure Data Collector 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*old_release*sbs.dat file in the *new_Software AG_directory*/infrastructuredc/bin/migrate directory. The settings tell the migration utility to migrate the configurations, data, and assets listed in " Infrastructure Data Collector Configurations, Data, and Assets that Can be Migrated" on page 85.

Migrate Using Custom or Default Imported Settings with Prompting

1. On the machine that hosts the new Infrastructure Data Collector, open a command window or shell, go to the *new_Software AG_directory*/infrastructuredc/bin/migrate directory, and run the command `migrate.{bat|sh}`.

2. The utility asks for the full path to the Software AG directory that contains the old Infrastructure Data Collector installation or to the ZIP file you made earlier.

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*old_release*sbs.dat file. For the migrate*old_release*sbs.dat file, specify *old_release* without periods (for example, 980, or 9100, or 1030).

Migrate Using Custom or Default Imported Settings without Prompting

On the machine that hosts the new Infrastructure Data Collector, open a command window or shell, go to the *new_Software AG_directory*/infrastructuredc/bin/migrate directory, and run the command below. For the migrate*old_release*sbs.dat file, specify the source release number without periods (for example, 980, or 9100, or 1030).

If an error occurs, the utility exits.

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

Run the Optimize Migration Utility

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

On each machine that hosts a new Optimize Analytic Engine or Web Services Data Collector, open a command window or shell, go to the *new_Software AG_directory*/common/migrate/Optimize/bin directory, and run the command below. For the migrate*old_release*sbs.dat file, specify *old_release* without periods (for example, 980, or 9100, or 1030).

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

Run the Application Platform Migration Utility

The Application Platform migration utility runs without prompting you for any information. If an error occurs, the utility exits.

On each machine that hosts a new Integration Server instance that hosts the Application Platform Support package or a new My webMethods Server that hosts Application Platform projects bundles, open a command window or shell, go to the *new_Software AG_directory*/common/migrate/AppPlatform/bin directory, and run the command below. For the

migrate*old_releases*bs.dat file, specify *old_release* without periods (for example, 980, or 9100, or 1030).

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

If you created web applications using WmTomcat on an Integration Server instance and you want to use those web applications with Application Platform in the new installation, but you did not have Application Platform on the old Integration Server instance, also specify the `-srcVersion old_release` parameter on the command.

Run the MashZone NextGen Migration Utility

The MashZone NextGen migration utility runs without prompting you for any information. If an error occurs, the utility exits.

On the machine that hosts the new MashZone NextGen, open a command window or shell, go to the *new_Software AG_directory*/MashZoneNG/prestocli/bin/migrate directory, and run the command below. For the migrate*old_releases*bs.dat file, specify *old_release* without periods (for example,1010, or 1030).

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Software AG_directory|ZIP_file}
-importFile {migrateold_releasesbs.dat|full_path_to_migrate.dat}
-silent true
```

Run the OneData Migration Utility

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

On the machine that hosts the new OneData, open a command window or shell, go to the *new_Software AG_directory*/profiles/ODE/bin/migrate directory, and run the command below.

```
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 new configuration data in the *new_Software AG_directory*/profiles/ODE/bin/onedata/config directory. The backup files have the extension .bck.

Run the Digital Event Services Migration Utility

The Digital Event Services migration utility runs without prompting you for any information. If an error occurs, the utility exits.

On the machine that hosts the new products, open a command window or shell, go to the *new_Software AG_directory*/common/migrate/DigitalEventServices/bin directory, and run the

command below. For the migrate*old_release*sbs.dat file, specify *old_release* without periods (for example, 9120).

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

23 Complete Final Upgrade Tasks for All Products

- Configure your new products. For instructions, see the product documentation for the new release.
- If you installed your new products on a different machine than your old products, make sure to update host names in your new 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 new products using the instructions in the product documentation. Also make sure any absolute paths in the new 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 product readmes for your old release+1 through the new release, including the readme for the Software AG Infrastructure. For example, if your old release is 9.9, read the information for releases 9.10 through 10.7. All readmes are available on the Software AG Documentation website. The product readmes contain this information:
  - Critical information and known and resolved issues for your products.
  - Changes to product behavior, services, parameters, properties, and APIs. Such changes can include additions, changes, deprecations, and removals. This information is especially important because you might need to modify product files or assets after migration to accommodate the changes.
- 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.
- After you have thoroughly tested your new environment, you can reclaim disk space by deleting ZIP files and files extracted from ZIP files from new machines, or by uninstalling products from old machines. To uninstall old products, use Command Central or the Software AG Uninstaller for the old release.