

# Upgrading Software AG Products On Premises

Version 10.15

October 2022

This document applies to webMethods Migration Framework 10.15 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.

Copyright © 2010-2024 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <https://softwareag.com/licenses/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

**Document ID: SAG-UPGR-1015-20240226**

# Table of Contents

<b>About this Guide</b> .....	7
Document Conventions.....	8
Online Information and Support.....	8
Data Protection.....	9
<b>1 Prepare for Upgrade</b> .....	<b>11</b>
Upgrades Covered in This Guide.....	12
Preparing, Planning, and Testing.....	12
Using Software AG Installer, Software AG Update Manager, Database Component Configurator, and Migration Utilities to Upgrade.....	12
Using Command Central to Upgrade.....	13
Documentation Needed to Perform the Upgrade.....	13
Troubleshooting.....	14
Support for Upgrade Across Operating Systems.....	15
General Upgrade Procedure Requirements.....	15
Upgrades and Migrations Covered in Other Guides.....	15
<b>2 Upgrade Software AG Infrastructure</b> .....	<b>17</b>
Install the New Product.....	18
Prepare the Old Installation.....	18
Migrate Software AG Infrastructure.....	19
Complete the Upgrade.....	21
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	21
<b>3 Upgrade Metering</b> .....	<b>23</b>
Install the New Products.....	24
Upgrade Metering Agent.....	24
Upgrade Metering Server.....	26
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	27
<b>4 Upgrade BigMemory Max or Terracotta</b> .....	<b>29</b>
Install the New Product.....	30
Prepare the Old Installation.....	30
Migrate BigMemory Max.....	31
Migrate Terracotta.....	31
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	33
<b>5 Upgrade Universal Messaging</b> .....	<b>35</b>
Universal Messaging Upgrade Guidelines.....	36

Install the New Product.....	37
Prepare the Old Installation.....	38
Migrate Universal Messaging.....	38
Complete the Upgrade.....	41
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	41
<b>6 Upgrade My webMethods Server.....</b>	<b>43</b>
Cluster Upgrade Guidelines.....	44
Install the New Product.....	44
Prepare the Old Installation.....	45
Migrate Database Components.....	46
Migrate My webMethods Server.....	46
Complete the My webMethods Server Upgrade.....	49
Complete the Business Rules Upgrade.....	50
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	51
<b>7 Upgrade Integration Server or Microservices Runtime and Hosted Wm Packages.....</b>	<b>53</b>
Cluster Upgrade Guidelines.....	54
Install the New Products.....	54
Prepare the Old Installation.....	55
Migrate Database Components.....	58
Migrate Integration Server or Microservices Runtime and Hosted Wm Packages.....	58
Complete the Integration Server Upgrade.....	63
Complete the Integration Server or Microservices Runtime Upgrade.....	64
Complete the Adapters or eStandards Modules Upgrade.....	65
Complete the ActiveTransfer Upgrade.....	65
Complete the CloudStreams Upgrade.....	65
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	67
<b>8 Upgrade Software AG Designer and Business Process Data.....</b>	<b>69</b>
Install the New Software AG Designer.....	70
Prepare the Old Software AG Designer.....	70
Migrate Database Components.....	71
Migrate Software AG Designer.....	71
Complete the Business Process Upgrade.....	76
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	77
<b>9 Upgrade Infrastructure Data Collector.....</b>	<b>79</b>
Install the New Product.....	80
Prepare the Old Installation.....	80
Migrate Infrastructure Data Collector.....	81
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	84

<b>10 Upgrade Optimize</b> .....	<b>85</b>
Install the New Product.....	86
Ways to Migrate.....	86
Prepare the Old Installation.....	87
Migrate Database Components.....	88
Configure and Start the Terracotta Server Array.....	88
Update Connections to Other Products.....	88
Migrate Optimize.....	89
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	92
<b>11 Upgrade the Asset Build Environment, Deployer, or Application Platform (Deprecated)</b> ...93	
Install the New Product.....	94
Upgrade the Asset Build Environment.....	94
Upgrade Deployer.....	95
Upgrade Application Platform (Deprecated).....	96
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	99
<b>12 Upgrade Apama</b> .....	<b>101</b>
Install the New Product.....	102
Prepare the Old Installation.....	102
Migrate Apama.....	103
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	103
<b>13 Upgrade Developer Portal, API Gateway, API Portal, or CentraSite</b> .....	<b>105</b>
API Gateway Cluster Upgrade Guidelines.....	106
Install the New Products.....	106
Upgrade API Gateway.....	107
Upgrade Developer Portal.....	108
Upgrade API Portal.....	110
Migrate API Portal 10.7 to Developer Portal 10.15.....	113
Upgrade CentraSite.....	115
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	118
<b>14 Upgrade MashZone NextGen</b> .....	<b>121</b>
Install the New Product.....	122
Upgrade from 10.1 or 10.3: Migrate MashZone NextGen.....	123
Upgrade from 9.10 or 9.12: Migrate MashZone NextGen.....	125
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	130
<b>15 Upgrade from 9.9: Upgrade Mobile Designer</b> .....	<b>131</b>
Install the New Product.....	132

Prepare the Old Installation.....	132
Migrate Mobile Designer.....	133
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	133
<b>16 Upgrade webMethods OneData.....</b>	<b>135</b>
Gather Information for Installation.....	136
Install the New Product.....	136
Prepare the Old Installation.....	137
Migrate Database Components.....	138
Migrate webMethods OneData.....	138
Complete the Upgrade.....	139
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	140
<b>17 Upgrade Zementis Predictive Analytics.....</b>	<b>143</b>
Install the New Product.....	144
Prepare the Old Installation.....	144
Migrate Zementis Predictive Analytics.....	145
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	145
<b>18 Upgrade Digital Event Services (Deprecated).....</b>	<b>147</b>
Digital Event Services Deprecation.....	148
Install the New Product.....	148
Prepare the Old Installation.....	148
Migrate Digital Event Services (Deprecated).....	149
Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup.....	150
<b>19 Migrate Database Components.....</b>	<b>153</b>
Requirements and Cautions.....	154
Prepare Database Components for Migration.....	154
Migrate Database Components Using Database Component Configurator.....	155
<b>20 Upgrade Using Command Central.....</b>	<b>159</b>
Supported Upgrade Scenarios.....	160
Upgrade Approaches.....	161
Upgrade Products in a Single Environment.....	161
Upgrade Products Using Software Stacks (10.1 and Later).....	163

# About this Guide

■ Document Conventions .....	8
■ Online Information and Support .....	8
■ Data Protection .....	9

---

This guide provides upgrade instructions for Software AG products.

## Document Conventions

---

Convention	Description
<b>Bold</b>	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	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

---

### Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.softwareag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.



---

## Product Training

You can find helpful product training material on our Learning Portal at <https://learn.softwareag.com>.

## Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://techcommunity.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://containers.softwareag.com/products> and discover additional Software AG resources.

## Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

## 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 Prepare for Upgrade

---

■ Upgrades Covered in This Guide .....	12
■ Preparing, Planning, and Testing .....	12
■ Using Software AG Installer, Software AG Update Manager, Database Component Configurator, and Migration Utilities to Upgrade .....	12
■ Using Command Central to Upgrade .....	13
■ Documentation Needed to Perform the Upgrade .....	13
■ Troubleshooting .....	14
■ Support for Upgrade Across Operating Systems .....	15
■ General Upgrade Procedure Requirements .....	15
■ Upgrades and Migrations Covered in Other Guides .....	15

## Upgrades Covered in This Guide

---

This guide explains how to upgrade the following products from 9.9 or later to 10.15:

- 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 10.15 installation to another 10.15 installation (for example, for a data center move).

This guide includes individual chapters for products that require product-specific tasks such as data migration as part of their upgrade. If no product-specific chapter exists for a product (for example, for ApplinX or EntireX), then you need only install the new product for the upgrade to be achieved.

Integration Agent does not exist in webMethods from version 10.11 onward. 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.

The products below diverged from general release numbering.

- In the webMethods 9.9, 9.10, and 9.12 releases, ActiveTransfer remained at its 9.8 release.
- In the webMethods 10.11 and 10.15 releases, API Portal remained at its 10.7 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), 10.11 (4.3.10), and 10.15 (4.4.0).

## 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.

## Using Software AG Installer, Software AG Update Manager, Database Component Configurator, and Migration Utilities to Upgrade

---

You can upgrade all products using Software AG Installer to install new products; Software AG Update Manager to install fixes; Database Component Configurator to migrate databases; and Software AG migration utilities to migrate product configurations, data, and assets. Migration utilities automatically migrate configurations and data from your old installation to your new

installation. For some products, migration utilities run without prompting you for any information (that is, silently). 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 `migrateold_releasesbs.dat`. The settings for default migrations are described in the product-specific chapters.

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.

Some products provide scripts instead of migration utilities for migrating configurations, data, and assets.

To use Software AG Installer, Software AG Update Manager, Database Component Configurator, migration utilities, and migration scripts, follow the instructions in this guide.

## Using Command Central to Upgrade

Command Central runs Software AG Installer, Software AG Update Manager, Database Component Configurator, and Migration Utilities to perform upgrades. For information about what products you can upgrade using Command Central, see [Software AG Command Central Feature Support Matrix](#).

Install and configure Command Central on a central machine and in each environment. Install the Command Central agent, Platform Manager, in each of your installations, and create repositories for products, fixes, and assets. For more information about upgrade options and instructions on how to perform them, see [“Upgrade Using Command Central” on page 159](#) in this guide.

## 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.

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 Gateway Administration*

- *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 Administration Guide*

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	<i>new_Software AG_directory/install/logs</i> directory
Product	In a logs directory in the product file structure, or in the <i>new_Software AG_directory/profiles/product/logs</i> directory.
Data migration	<i>new_Software AG_directory/install/logs</i> directory and product directories indicated by the upgrade procedures
Database migration	<i>new_Software AG_directory/common/db/logs</i> 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.

## Support for Upgrade Across Operating Systems

Integration Server, Microservices Runtime, webMethods 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).

## 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 this guide 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.
- When you install the new products, you also 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.

## 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 <i>Supported Upgrade Paths</i> .
webMethods adapters	See the adapter product documentation.

Upgrade	Information
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	<p>webMethods Broker has been deprecated.</p> <p>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 <i>Migrating from webMethods Broker to Software AG Universal Messaging</i>.</p> <p>If you want to stay on webMethods Broker, first upgrade webMethods Broker as instructed in <i>Installing and Upgrading webMethods Broker</i>, and then upgrade your other products as instructed in this guide.</p>
Command Central	See the Command Central product documentation.
webMethods eStandards Modules	See the eStandards Module product documentation.
Mediator to API Gateway	See the API Gateway product documentation.
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.



## 2 Upgrade Software AG Infrastructure

---

■ Install the New Product .....	18
■ Prepare the Old Installation .....	18
■ Migrate Software AG Infrastructure .....	19
■ Complete the Upgrade .....	21
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	21

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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 product on a Windows system, and you installed it as a Windows service, the default startup mode for the service is Automatic. To prevent the new product from starting accidentally before this procedure instructs you to start it, set the service to Manual. If you installed on a UNIX system, and you have scripts that automatically start daemons, disable the scripts for the same reason.

## Prepare the Old Installation

---

1. Shut down the products in the old installation.

**If the product is running**

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG product release.

---

**If the product is running**

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

2. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level jvm directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.
  - a. You can reduce the ZIP file size by first moving the log files out of the *old\_Software AG\_directory/profiles/CTP/logs* directory.
  - b. On the product's old machine, open a command window or shell.
  - c. Go to the Software AG directory that contains the old product and enter this command:
 

```
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.
  - d. 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.

## 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.

### 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.

## Run the Software AG Infrastructure Migration Utility

This migration utility runs without prompting for any information. If an error occurs, 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.

1. Make sure none of the new products are running.
2. 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 `migrateold_releasesbs.dat` file, specify `old_release` without periods (for example, 990, or 9100, or 1030).

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Software_AG_directory|ZIP_file}
```

```
-importFile migrateold_releasesbs.dat
-silent true
```

3. 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.

## Complete the 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.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 3 Upgrade Metering

---

- Install the New Products ..... 24
- Upgrade Metering Agent ..... 24
- Upgrade Metering Server ..... 26
- Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup ..... 27

## Install the New Products

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

If you installed your new product on a Windows system, and you installed it as a Windows service, the default startup mode for the service is Automatic. To prevent the new product from starting accidentally before this procedure instructs you to start it, set the service to Manual. If you installed on a UNIX system, and you have scripts that automatically start daemons, disable the scripts for the same reason.

## Upgrade Metering Agent

---

### Prepare the Old Installation

1. Shut down all of the old products that used Metering Agent.

**If the product is running**

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG product release.

---



**If the product is running**

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

2. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level jvm directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.
  - a. On the product's old machine, open a command window or shell.
  - b. Go to the Software AG directory that contains the old product and enter this command:
 

```
jar cfm ZIP_file_name.zip common/metering install/products
```
  - c. 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.

## Migrate Metering Agent

### Configurations that will be migrated

- Metering Agent configuration file.

### Cached measurements that will be migrated

- The Metering Agent cache files stored in the *SotwareaAG\_Directory/common/metering/storage* directory.

### Run the Metering Agent Migration Utility

This migration utility runs without prompting for any information. If an error occurs, 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.

1. Make sure none of the new products are running.
2. On the machine that hosts the new products, open a command window or shell, go to the `new_Software AG_directory/common/metering/migrate` directory, and run the command below. For the `migrateold_releasesbs.dat` file, specify `old_release` without periods (for example, 1070).

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

## Upgrade Metering Server

### Prepare the Old Installation

1. Shut down all of the old products that used Metering Server.

#### If the product is running Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

2. If your old and new installations are on different machines, and the Metering Server database is in the default location, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level `jvm` directory name includes additional information, such as `/jvm/jvm160_32`, or `/jvm/jvm170`, or `/jvm/jvm_64`.

#### Note:

If the Metering Server database is not in the default location, contact Software AG Global Support for assistance.

- a. On the product's old machine, open a command window or shell.
- b. Go to the Software AG directory that contains the old product and enter this command:

```
jar cfM ZIP_file_name.zip profiles/CTP/database profiles/CTP/configuration
install/products
```

- c. 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.

## Migrate the Metering Server

### Folders and Locations that Will be Migrated

- Metering Server database.

### Run the Metering Server Migration Utility

This migration utility runs without prompting for any information. If an error occurs, 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.

1. Make sure none of the new products are running.
2. On the machine that hosts the new products, open a command window or shell, go to the *new\_Software AG\_directory/common/metering-server/migrate* directory, and run the command below. For the *migrateold\_releasesbs.dat* file, specify *old\_release* without periods (for example, 1070).

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

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:

- Update host names in the new product and in the product database tables.
- Specify the correct host names when you configure the new product and when you set up connections to other products.
- 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 4 Upgrade BigMemory Max or Terracotta

---

■ Install the New Product .....	30
■ Prepare the Old Installation .....	30
■ Migrate BigMemory Max .....	31
■ Migrate Terracotta .....	31
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	33

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Installation

---

1. Shut down the old installation.

**If the product is running** Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

2. Shut down all non-webMethods clients.

## Migrate BigMemory Max

**Note:**

Terracotta BigMemory Max was renamed BigMemory Max in release 10.5.

This section explains how to migrate BigMemory 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.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.



# 5 Upgrade Universal Messaging

---

■ Universal Messaging Upgrade Guidelines .....	36
■ Install the New Product .....	37
■ Prepare the Old Installation .....	38
■ Migrate Universal Messaging .....	38
■ Complete the Upgrade .....	41
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	41

## Universal Messaging Upgrade Guidelines

---

When you install the new Universal Messaging on a new machine, provide the same configuration and the same or increased resources on the target machine. For example, make sure to allocate the same or more memory capacity and the same file descriptors.

## Universal Messaging Cluster Upgrade Guidelines

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 server installations, create a cluster that matches your old cluster, for example, a two-node or a three-node cluster. You can install the new Universal Messaging server installations and the latest fixes on them in parallel.
- When you install the latest fixes on the new Universal Messaging server installations, 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. The migration utility creates the new server instances in the new installations and migrates their data.
- 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 the *Universal Messaging Administration Guide*.

## Install the New Product

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

On the product installer panel or in response to a prompt, select **Side-by-side installation for upgrade**. No instance will be created.

### Note:

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Installation

---

1. Install the latest product fixes on the old Universal Messaging, then start the old Universal Messaging.
2. Shut down the old installation.

### If the product is running Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

---

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

---

On a UNIX system Use the shut down instructions in the product documentation for your old release.

---

3. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level jvm directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.
  - a. On the product's old machine, open a command window or shell.
  - b. Go to the Software AG directory that contains the old product and enter this command:

```
jar cfM ZIP_file_name.zip install/products UniversalMessaging/server
```
  - c. 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.

## Migrate Universal Messaging

---

### Configurations, Data, and Assets that Will be Migrated

- Server instances.
- Server instance data directories.

Universal Messaging 10.11 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 10.11 server.

## Run the Migration Utility

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

If you run the migration 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.

## 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 `migrateold_releasesbs.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 “[Configurations, Data, and Assets that Will be Migrated](#)” on page 38.

### 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 `migrateold_releasesbs.dat` file, specify `old_release` without periods (for example, 990 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
```



## Complete the 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/serverinstance\_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.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 6 Upgrade My webMethods Server

---

■ Cluster Upgrade Guidelines .....	44
■ Install the New Product .....	44
■ Prepare the Old Installation .....	45
■ Migrate Database Components .....	46
■ Migrate My webMethods Server .....	46
■ Complete the My webMethods Server Upgrade .....	49
■ Complete the Business Rules Upgrade .....	50
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	51

## Cluster Upgrade Guidelines

---

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 rest of this chapter. You must perform all documented tasks for your products in the order they are written in this chapter, with these additional tasks or exceptions:

- 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*.

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

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.

On the product installer panel or in response to a prompt, select **Side-by-side installation for upgrade**. No instance will be created.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

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

## Prepare the Old Installation

1. Install the latest product fixes on the old My webMethods Server, then start the old My webMethods Server.
2. 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, make the changes in that file, and 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 (see *Administering My webMethods Server*). 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.

3. Shut down the old installation.

**If the product is running**

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

4. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source.
  - a. 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 below. The command creates a ZIP file named *mws.zip* in *old\_Software AG\_directory/MWS/bin/migrate* directory.

```
ZIP-mws. {bat | sh}
```

- b. 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.

---

## Migrate Database Components

For instructions on migrating database components for My webMethods Server, see [“Migrate Database Components” on page 153](#).

---

## Migrate My webMethods Server

### Instances, Configurations, Data, and Assets that Can be Migrated

You can choose to migrate the following:

- Server instances, with this exception: 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.

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

After migrating, the utility deletes old configuration files, and deletes old properties that are not used by the new My webMethods Server from the new configuration files. The utility also synchronizes the My webMethods Server installation directory with the database.

## Run the Migration Utility

If you run the migration 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.

## 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 `migrateold_releasesbs.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 *migrateold\_releasesbs.dat* file, specify *old\_release* without periods (for example, 990, 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
```

## 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 Migrated Database to Live Migrated Database

If you used a clone migrated database to test your new installation (recommended), and you later want to use your live migrated 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 the 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."

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 7 Upgrade Integration Server or Microservices Runtime and Hosted Wm Packages

---

■ Cluster Upgrade Guidelines .....	54
■ Install the New Products .....	54
■ Prepare the Old Installation .....	55
■ Migrate Database Components .....	58
■ Migrate Integration Server or Microservices Runtime and Hosted Wm Packages .....	58
■ Complete the Integration Server Upgrade .....	63
■ Complete the Integration Server or Microservices Runtime Upgrade .....	64
■ Complete the Adapters or eStandards Modules Upgrade .....	65
■ Complete the ActiveTransfer Upgrade .....	65
■ Complete the CloudStreams Upgrade .....	65
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	67

## Cluster Upgrade Guidelines

---

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.

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 rest of this chapter. You must perform all documented tasks for your products in the order they are written in this chapter, with these additional tasks or exceptions:

- 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 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 Integration Servers or 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*.

## Install the New Products

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

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, for Integration Server, business process packages that were generated by users from Software AG Designer.

On the Integration Server or Microservices Runtime installer panel or in response to a prompt, point the database connection at the ISInternal database component you want to use with the new installation.

On the ActiveTransfer Server installer panel or in response to a prompt, point the database connection at the ActiveTransfer database component you want to use with the new ActiveTransfer Server.

On the Trading Networks installer panel or in response to a prompt, point the database connection at the TradingNetworks database component you want to use with the new Trading Networks.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Installation

---

1. Start and connect products.
  - a. Start the old Integration Server or Microservices Runtime and open the old Integration Server Administrator or Microservices Runtime Administrator.
  - b. 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.
2. 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.

3. Suspend triggers and drain queues.
  - a. Go to the **Server > 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*.
  - b. Go to the **Messaging > webMethods triggers** 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.
  - c. Go to the **Messaging > webMethods settings** page. Make sure the **CSQ Count** field shows 0 for the Universal Messaging connection alias.
  - d. Go to the **Messaging > JMS settings** page. In the **JMS Connection Alias Definitions** area, make sure the **CSQ Count** field shows 0 for every JMS connection alias.
4. This guide refers to different types of packages for Integration Server or Microservices Runtime. You might need to adjust your custom packages.

- 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.
- Custom packages. These include Integration Server or Microservices Runtime packages created by users in Software AG Designer and business process packages generated by Integration Server users from Software AG Designer. The migration utility will scan the old installation for custom packages. However, it will not find custom packages whose names start with Wm, as this naming convention is used for packages provided by Software AG. If you have custom packages whose names start with Wm, and you want to migrate them, go to the *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 *Wmname* for custom packages.

5. Upgrade ActiveTransfer with the following considerations:



- When you migrate ActiveTransfer from version 10.11 or lower to version 10.15, you must also migrate My webMethods Server to version 10.15. For more information about how to migrate My webMethods Server, see “[Migrate My webMethods Server](#)” on page 46.
- For ActiveTransfer versions 9.9, 9.10, or 9.12, install Fix 11.

**Note:**

The ActiveTransfer release number in versions 9.9, 9.10, or 9.12, releases was 9.8.

6. If the old Integration Server or Microservices Runtime hosts a Process Engine, install the latest product fixes on the old Process Engine, then start the old Process Engine.
7. Shut down the old installation.

**If the product is running Take this action...**

As a Windows service	Shut down from the Windows Services window. Services are listed as <i>Software AG product release</i> .
----------------------	---

As a Windows application	Shut down from the Windows start menu. Applications are listed as <b>Software AG &gt; Stop Servers &gt; product</b> .
--------------------------	---

On a UNIX system	Use the shut down instructions in the product documentation for your old release.
------------------	---

**Note:**

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

8. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level jvm directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.
  - a. For 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.

Upgrade from 10.3 or earlier: For Microservices Runtime, 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. 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.

- b. On the product's old machine, open a command window or shell.
- c. Go to the Software AG directory that contains the old product and enter this command:  

```
jar cfM ZIP_file_name.zip *
```
- d. 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.

## Migrate Database Components

---

For instructions on migrating database components for Integration Server or Microservices Runtime and hosted Wm packages, see [“Migrate Database Components” on page 153](#).

## Migrate Integration Server or Microservices Runtime and Hosted Wm Packages

---

### Configurations, Data, and Assets that Can or Will be Migrated

You can choose whether to migrate the following:

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

The utility will also do the following:

- If you are upgrading CloudStreams, migrate configuration artifacts related to administering CloudStreams Server.
- If the old installation used the embedded database, copy database tables from the old installation to the new installation and upgrade the tables to the new format, if the format has changed.

- Update the SERVER ID stored in database tables and configuration files to reflect the new host machine.
- Update keystore aliases of type PKCS12 whose keystore provider name is SunJSSE to use Bouncy Castle as a provider.
- Add a new property named "Validate schemas using Xerces" to existing Web service descriptors, and set the new property 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 more information, see *webMethods Service Development Help*.
- If you are migrating Business Rules data, upgrade Business Rule projects. You might see XML parsing messages due to a Java bug; you can ignore these messages.

After the migration, the utility deletes old configuration files, and deletes old properties that are not used by the new installation from the new configuration files.

## Run the Migration Utility

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.

If you are upgrading an Integration Server or Microservices Runtime that hosts Deployer, follow the instructions in [“Upgrade the Asset Build Environment, Deployer, or Application Platform \(Deprecated\)”](#) on page 93.

If you run the migration 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.

## Perform a Custom Migration

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.

For Integration Server, 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 selected packages only, the utility lists each of the packages above and asks whether to migrate it. If you choose to migrate all custom packages, the utility also migrates the `WmChemPayloads`, `WmPapinetPayloads`, `WmRNPIps`, `WmFINMessages`, and `WmFIXMessages` packages, if present, for Integration Server. These packages contain IS documents for the corresponding eStandards Modules and the schemas for those documents.:
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 or 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.

**Important:**

During installation of the new Integration Server or Microservices Runtime, you specified a custom password as the default for the product. If you want to use this custom password, choose to migrate selected configuration file only, and do not migrate the `users.cnf` file. If you migrate `users.cnf`, the new customer password will not be available, and instead the password you used with the old Integration Server or Microservices Runtime will be in effect for the new product.

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 or 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 Integration Server or 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. 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 `TransactionLateFADData`.
14. For Integration Server, the utility asks whether to migrate JAAS configurations.
15. for Integration Server, 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.

## 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 `migrateold_releasesbs.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 Migration Utility” on page 59](#). 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 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.

for Integration Server, 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.

- The utility asks whether to import migration settings. Enter Y and, when prompted, provide the full path to the migrate.dat file or the migrateold\_releasesbs.dat file. For the migrateold\_releasesbs.dat file, specify *old\_release* without periods (for example, 990, or 9100, or 1030).

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

If you are using a cloned database, 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 migrateold\_releasesbs.dat file, specify the source release number without periods (for example, 990, 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
```

## 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 connect 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.
- If you want to configure ActiveTransfer to automatically store certificates into a database, run the `wm.mft.admin:updateFileBasedCertToDB` service. (If you later want to revert this behavior, run the `wm.mft.admin:updateCertFromDBToFile` service.)

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
<code>allPackages</code>	<p><b>Boolean string.</b> Optional. Set to:</p> <ul style="list-style-type: none"> <li>■ <code>true</code> to migrate CloudStreams artifacts from all custom packages that are dependent on <code>WmCloudStreams</code>.</li> </ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>■ <code>false</code> to migrate CloudStreams artifacts from only custom packages specified on the <code>packages</code> parameter. This is the default.</li> </ul>
<code>packages</code>	<b>Object.</b> 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
<code>packageName</code>	<b>String.</b> Name of the custom package.
<code>success</code>	<b>String.</b> Value that indicates whether migration succeeded ( <code>true</code> ) or failed ( <code>false</code> ).
<code>message</code>	<b>String.</b> Information about the migration, such as number of CloudStreams artifacts found in the package and number of CloudStreams artifacts that were successfully migrated.
<code>info</code>	<b>Object.</b> Optional. If the service issued info messages during the migration, string array of the messages.
<code>errors</code>	<b>Object.</b> 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.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.



# 8 Upgrade Software AG Designer and Business

## Process Data

---

■ Install the New Software AG Designer .....	70
■ Prepare the Old Software AG Designer .....	70
■ Migrate Database Components .....	71
■ Migrate Software AG Designer .....	71
■ Complete the Business Process Upgrade .....	76
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	77

## Install the New Software AG Designer

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

If you are upgrading products involving Software AG Designer projects, make sure the target machine has enough space for the projects you will migrate.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Software AG Designer

---

1. Export Integration Server or Microservices Runtime definitions.
  - a. 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**.

- b. Click **Export** and complete the dialog box. Software AG Designer will save the file with the extension `.properties`.
2. Export CloudStreams server definitions and projects.
  - a. 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`.
  - b. Open the CloudStreams Development perspective and go to the CloudStreams Governance view. Right-click a governance project to export, click **Export**, and complete the wizard. The project folder will be exported.
  - c. Go to 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).
3. Export preferences.
  - a. In the old Software AG Designer, open the **File > Export** wizard. In the Select panel, go to **General > Preferences** and click **Next**.
  - b. 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`.
4. Shut down the old installation.

#### If the product is running Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

## Migrate Database Components

For instructions on migrating database components for Software AG Designer, see [“Migrate Database Components” on page 153](#).

## 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, `workspace1015`).

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 4.19, 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 Zulu Java 11 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(Deprecated), 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 (Deprecated), 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 <b>Problems</b> tab, click each error and resolve it.
Application Platform (Deprecated)	All projects compile without errors. If errors appear in the <b>Problems</b> 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 <b>Solutions</b> tab under the Processes node.
Business Rules	All imported business rules projects appear in the <b>Solutions</b> 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**.**Note:**

The **Upgrade Project** button appears only when an update is available for the rule project. If the **Upgrade Project** option is not visible, you can use the rule project in the currently installed version of Software AG Designer as it is.

## 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 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 (Deprecated), 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 (Deprecated), you will see a `/DigitalEventServices` directory. Erase the custom configuration and re-create it from scratch using Digital Event Services (Deprecated) 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 **Messaging > webMethods triggers** 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*.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 9 Upgrade Infrastructure Data Collector

---

■ Install the New Product .....	80
■ Prepare the Old Installation .....	80
■ Migrate Infrastructure Data Collector .....	81
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	84

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Installation

---

1. Install the latest product fixes on the old Infrastructure Data Collector, then start the old or Infrastructure Data Collector.
2. Shut down the old installation.



**If the product is running** Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

3. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level jvm directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.
  - a. You can reduce the ZIP file size by first moving the log files out of the *old\_Software AG\_directory/profiles/InfraDC/logs* directory.
  - b. On the product's old machine, open a command window or shell.
  - c. Go to the Software AG directory that contains the old product and enter this command:
 

```
jar cfm ZIP_file_name.zip common/conf install/products profiles/InfraDC
```
  - d. 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.

## Migrate Infrastructure Data Collector

### Configurations, Data, and Assets that Can be Migrated

You can choose to migrate the following:

- 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).

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.

## Run the Migration Utility

If you run the migration 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.

## 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 migrateold\_releasesbs.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 .

### 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 migrateold\_releasesbs.dat file. For the migrateold\_releasesbs.dat file, specify *old\_release* without periods (for example, 990, 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 migrateold\_releasesbs.dat file, specify the source release number without periods (for example, 990, 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
```

## Complete the 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**.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 10 Upgrade Optimize

---

■ Install the New Product .....	86
■ Ways to Migrate .....	86
■ Prepare the Old Installation .....	87
■ Migrate Database Components .....	88
■ Configure and Start the Terracotta Server Array .....	88
■ Update Connections to Other Products .....	88
■ Migrate Optimize .....	89
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	92

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Ways to Migrate

---

You can migrate from your old Optimize installation using one of these methods:

- Upgrade from 9.12 or later: Use the migration utility to migrate your Analytic Engine and Web Service Data Collector configurations from your old Optimize installation to the new one.
- Upgrade from 9.9 or later: Migrate your database components using the Database Component Configurator, and then migrate your Central Configurator System (CCS) environments from the Define Environments page in My webMethods Server.

- Upgrade from 9.9 or later: Export your CCS environments from your old Optimize installation, and then import them to your new Optimize installation from the Define Environments page in My webMethods Server.

## Prepare the Old Installation

1. Install the latest product fixes on the old Optimize, then start the old Optimize.
2. If you are going to reuse your old CCS environments by importing them into your new Optimize installation, you must first export them from the old Optimize installation. Open My webMethods and go to the **Applications > Administration > System-Wide > Environments > Define Environments** page. Select your CCS environments and click **Export** to save them to a local file.
3. Shut down the old installation.

### If the product is running

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

### Note:

It is especially important to shut down all old Optimize instances (for example, Analytic Engines) that connect to database components you are going to migrate.

4. Upgrade from 10.7 or later: If you are going to use the migration utility, and your old and new Optimize installations are on different machines, create a ZIP file of the old product installation to use as the migration source.
  - a. 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 below. 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.

```
ZIP-optimize.{bat|sh}
```

### 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.

- b. 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 below. 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.

```
ZIP-dc.{bat|sh}
```

**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.

## Migrate Database Components

---

For instructions on migrating database components for Optimize, see [“Migrate Database Components” on page 153](#).



## 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

---

1. Update the connection to Process Engine.
  - a. Open Integration Server Administrator and point to a new Integration Server that hosts a Process Engine.
  - b. Go to the **Packages > Management** page and click  for the WmPRT package.
  - c. Click **Settings** in the left navigation bar and then click **Edit Process Engine Settings**.
  - d. In the **JMS Server URL** field, identify the Universal Messaging host name and port as `nsp://host:port`. The default port is 9000.
  - e. Click **Submit** and then reload the WmPRT package.
  - f. Repeat these steps for every new Integration Server that hosts a Process Engine.
2. Update the connection to the Optimize support package on Integration Server.
  - a. Open Integration Server Administrator or Microservices Runtime Administrator and point to a new Integration Server or Microservices Runtime that hosts an Optimize Support package.
  - b. Go to the **Packages > Management** page and click  for the WmOptimize package.
  - c. Identify the Analytic Engine host machine and port. The default port is 12503.



- d. In the **JMS Server URL** field, identify the Universal Messaging host name and port as `nsp://host:port`. The default port is 9000.
- e. Click **Submit** and then reload the WmOptimize package.
- f. Repeat these steps for every Integration Server or Microservices Runtime that hosts an Optimize Support package.

## Migrate Optimize

### Configurations, Data, and Assets that Can be Migrated

You can choose to migrate the following:

- 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.

### Upgrade from 9.12 or Later: Migrate Using the Migration Utility

This migration utility runs without prompting for any information. If an error occurs, the utility exits. The utility does not revert the new product installation. You can address the issue and rerun the utility.

On each machine that hosts a new Optimize Analytic Engine or Web Service Data Collector, open a command window or shell, go to the `new_Software AG_directory/optimize/analysis/bin/migrate` directory or `new_Software AG_directory/optimize/dataCollector/bin/migrate` directory, and run the command below. For the `migrateold_releasesbs.dat` file, specify `old_release` without periods (for example, 912, or 1030).

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

#### Note:

Since the migration utility does not interact with the Central Configurator System database used by My webMethods Server, your migrated configuration will be visible only in Command Central.

If your new Optimize installation is located on a different machine than your old Optimize installation, you will need to update your Analytic Engine and Web Service Data Collector configurations. See the *Configuring BAM* guide for detailed instructions on how to configure Analytic Engine and Web Service Data Collector from Command Central.

## Upgrade from 9.9 or Later: Migrate Using the Define Environments Page in My webMethods Server

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

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

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 CCS Environment Definition

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

1. Start the new My webMethods Server and open My webMethods.
2. Go to the **Applications > Administration > System-Wide > Environments > Define Environments** page.
  - a. If you have migrated the database components from your old Optimize installation to your new Optimize installation, the **Define Environments** page will display your old CCS environments. To migrate these environments, select them and click **Migrate Environment**.
  - b. If you have configured your new Optimize installation to use a clean database, the **Define Environments** page will be empty. Click **Import Environment** and select the file to which you exported the CCS environments from your old installation. When prompted, confirm the migration.

**Note:**

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

**Note:**

The environments are migrated in their current state. If they are unconfigured in the old installation, they will remain unconfigured in the new installation.

3. 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**.
4. Click the **Define Environments** tab, click the environment to deploy, modify the configuration if necessary, and then click **Finish**.
5. Go to *Deploy the New Optimize Environment*, below.

## Deploy the New Optimize Environment

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

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`.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 11 Upgrade the Asset Build Environment, Deployer, or Application Platform (Deprecated)

---

■ Install the New Product .....	94
■ Upgrade the Asset Build Environment .....	94
■ Upgrade Deployer .....	95
■ Upgrade Application Platform (Deprecated) .....	96
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	99

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

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.

## Upgrade 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.

## Upgrade Deployer

### Prepare the Old Installation

1. 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*.
2. Shut down the old installation.

#### If the product is running Take this action...

As a Windows service	Shut down from the Windows Services window. Services are listed as Software AG <i>product release</i> .
----------------------	---

As a Windows application	Shut down from the Windows start menu. Applications are listed as <b>Software AG &gt; Stop Servers &gt; product</b> .
--------------------------	---

On a UNIX system	Use the shut down instructions in the product documentation for your old release.
------------------	---

3. 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.

## 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.

### Run the Migration Utility

Migrate Deployer and the host Integration Server or Microservices Runtime using the Integration Server migration utility (see [“Upgrade Integration Server or Microservices Runtime and Hosted Wm Packages” on page 53](#)).

## Upgrade Application Platform (Deprecated)

### Prepare the Old Installation

1. Shut down the old installation.

#### If the product is running

As a Windows service Shut down from the Windows Services window. Services are listed as Software  
*AG product release.*

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software**  
**AG > Stop Servers > product .**

On a UNIX system Use the shut down instructions in the product documentation for your old  
 release.

2. If the new and old My webMethods Servers that host Application Platform projects bundles are on the same machine, also shut down the old My webMethods Servers.
3. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level jvm directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.
  - a. 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).
  - b. On the product's old machine, open a command window or shell.
  - c. Go to the Software AG directory that contains the old product and enter this command:
 

```
jar cfm ZIP_file_name.zip install/products profiles
```
  - d. 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.

## Migrate 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.

## Run the Migration Utility

This migration utility runs without prompting for any information. If an error occurs, the utility exits. The utility does not revert the new product installation. You can address the issue and rerun the utility.

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 *migrateold\_releasesbs.dat* file, specify *old\_release* without periods (for example, 990, 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.

## Complete the 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.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.

- If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
- You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 12 Upgrade Apama

---

■ Install the New Product .....	102
■ Prepare the Old Installation .....	102
■ Migrate Apama .....	103
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	103

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

On the product installer panel or in response to a prompt, if you are 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.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Installation

---

Shut down the old installation.

### If the product is running... Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

## 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 that generated artifacts such as Apama 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.

#### Note:

Command Central integration with Apama is deprecated and will be removed in a later release.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.

- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.



# 13 Upgrade Developer Portal, API Gateway, API Portal, or CentraSite

---

■ API Gateway Cluster Upgrade Guidelines .....	106
■ Install the New Products .....	106
■ Upgrade API Gateway .....	107
■ Upgrade Developer Portal .....	108
■ Upgrade API Portal .....	110
■ Migrate API Portal 10.7 to Developer Portal 10.15 .....	113
■ Upgrade CentraSite .....	115
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	118

## API Gateway Cluster Upgrade Guidelines

---

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 rest of this chapter. You must perform all documented tasks for your products in the order they are written in this chapter, with these additional tasks or exceptions:

- 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 the *webMethods API Gateway Administration* guide.

## Install the New Products

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

Upgrade from 10.7: In the webMethods 10.15 release, API Portal remained at its 10.7 release. You can use your existing API Portal 10.7, you can install API Portal 10.7 on a new machine if necessary, or you can migrate to Developer Portal 10.15.

For CentraSite, if you want to install the 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.

### Note:

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Upgrade API Gateway

### Prepare the Old Installation

1. Install the latest product fixes on the old API Gateway, then start the old API Gateway.
2. Shut down the old installation.

**If the product is running** Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

## Migrate Database Components

For instructions on migrating database components for API Gateway, see [“Migrate Database Components” on page 153](#).

## Migrate 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.

### Perform the Migration

For instructions on upgrading API Gateway, see the *webMethods API Gateway Upgrade and Migration* guide.

## Upgrade Developer Portal

---

### Note:

After you migrate Developer Portal to version 10.15, you have to manually move any customizations you have made to the `dpo_wrapper.conf` file and any file-level changes of your old Developer Portal installation to your new installation.

## Prepare the Old Installation

1. Install the latest product fixes on the old Developer Portal and start it.

2. Open Developer Portal in a browser and log on with your Administrator credentials.
3. Go to **Administration > Backup and Restore**, select the required modules, and click **Backup**.  
Developer Portal creates a zip file that contains the backup data.
4. Shutdown old Developer Portal, if both old and new Developer Portal installations are on the same machine:
  - a. To stop the running data store instance, click on the **Start** menu, expand the Developer Portal menu group, and click **Stop API Gateway Data Store 10.11**.
  - b. Stop Software AG Runtime 10.11 services.

## Assets that are Migrated

- APIs (REST, SOAP, ODATA, HYBRID)
- Plans & Packages
- Applications
- Providers
- Teams & Communities
- Users & groups
- UMC configurations
- Themes
- Analytics data

## Perform the Migration

1. If the new and old Developer Portal installations are on the same machine, make sure the old Developer Portal is shutdown.
2. Ensure that Software AG Runtime 10.15 service is running and then to start the data store instance click on the **Start** menu, expand the Developer Portal menu group, and click **Start API Data Store 10.15**.
3. Open Developer Portal in a browser and log on with your Administrator credentials.
4. Go to **Administration > Backup and Restore**, click **Restore**, and then select the modules you backed up earlier.
5. Upload the backup and click **Restore**.

## Upgrade API Portal

---

### Upgrade from 9.9 through 10.5: Prepare the Old Installation

1. You must have the functional privileges below to upgrade API Portal. 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.

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

2. Install the latest product fixes on the old API Portal, then start the old API Portal.
3. Start the old API Portal Cloud Controller.
4. 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
```

5. Run this command:

```
acc> startall
```

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

```
acc> list
```

7. 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
```

8. 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 acc> backup tenant tenant_name to full_path_to_backup_file
10.5 username=your_user_name password=your_password
```

9. 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.
10. Shut down the old 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
```

## Migrate API Portal

In the webMethods 10.15 release, API Portal remained at its 10.7 release. If you are using your existing API Portal 10.7 with the webMethods 10.15 release, skip this section.

For all other upgrade paths, follow the steps in this section to migrate your old assets to the new installation.

### 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.

## Perform the Migration

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	<code>http://load_balancer_host:load_balancer_port/acc/rest/nodes/</code>
Method type	POST
Content type	application/json
Payload	<code>{"nodename": "host_name", "hostname": "host_name", "port": "cloud_agent_port", "username": "Clous", "password": "g3h31m"}</code>

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.



Old release	Run these commands...
9.9 9.10 9.12	<pre>acc&gt; set acc config backup.restore.tenant.app.types = UMC, ABS, ADS, ECP acc&gt; restore tenant <i>tenant_name</i> from <i>full_path_to_backup_file</i> username=<i>your_user_name</i> password=<i>your_password</i> acc&gt; invoke prepareTenant on <i>bundle_runnable_name</i> tenant.name=<i>tenant_name</i></pre>
10.1 10.3 10.4 10.5	<pre>acc&gt; restore tenant <i>tenant_name</i> from <i>full_path_to_backup_file</i> username=<i>your_user_name</i> password=<i>your_password</i> acc&gt; invoke prepareTenant on <i>bundle_runnable_name</i> tenant.name=<i>tenant_name</i></pre>

## Complete the 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 API Portal 10.7 to Developer Portal 10.15

In the webMethods 10.15 release, API Portal remained at its 10.7 release. You can migrate from API Portal 10.7 to Developer Portal 10.15.

### Prepare the Old Installation

1. Install the latest product fixes on the old API Portal, then start the old API Portal.
2. If any user registrations are waiting to be approved, complete the approvals in API Portal.
3. Create a backup of the old API Portal installation.
  - a. Trigger the backup by invoking the REST endpoint as Administrator. The command below shows how to back up all modules. If you only want to migrate certain modules, specify only those modules on the command.

```
POST http://<host>:<port>/abs/apirepository/migration/backup
{
  "modules": ["User","Core","Collaboration","Analytic_Metrics","Analytic_Events"]
}
```

You will see a progress "id" handle such as the following:

```
{
  "id": "fb0fda97-996d-4163-bbb0-b1ce6a7a2090",
  "_self":
  "/abs/apirepository/migration/status/fb0fda97-996d-4163-bbb0-b1ce6a7a2090"
```

```
}
```

- b. Check the status of the backup by entering this command:

```
GET http://<host>:<port>/abs/apirepository/migration/status/{progressId}
```

When the status returned is **SUCCEEDED**, download the backup archive file using the REST endpoint as shown below, where *progressID* is the "id" value from the SUCCEEDED status message..

```
GET http://<host>:<port>/abs/apirepository/migration/status/progressID/backup
```

## Migrate API Portal to Developer Portal

### Assets that Can be Migrated

- APIs (REST, SOAP, ODATA, HYBRID) .
- Packages.
- Applications.
- Providers.
- Communities and teams.
- Users and groups.
- UMC configurations (LDAP, SMTP, Policies, MFA) .
- Transactions events and transactional logs for APIs (analytics).
- API followers and ratings (collaboration)

### Limitations

- You cannot migrate UI themes, or collaboration data such as topics and comments.
- If you have any private communities that include users with the API consumer role as the Community administrator, these users will not have Community administrator rights after the migration.
- Client SDK download support is not available.
- If "system" user is part of the followers of an API, then this data is not migrated.

### Perform the Migration

1. Start the new Developer Portal and login as a user with Administrator privileges.

2. Go to **Administration > Backup and Restore**, click **Restore**, and then select the modules you backed up earlier.
3. Upload the backup and then click **Restore**.

## Complete the Migration

Update OAUTH applications as follows:

1. Log on to Developer Portal and go to the **Applications** page.
2. Click the edit icon next to an OAUTH application and go to the **Edit** page.
3. In the **Associated Callbacks** field, delete the callback URL.
4. In the **Callbacks** field, type the URL  
`https://host_name:https_port/portal/rest/v1/oauth/callback` and click **Add**.
5. Click **Save**.
6. Repeat for every OAUTH application.

## Upgrade CentraSite

---

### Prepare the Old Installation

1. Install the latest product fixes on the old CentraSite, then start the old CentraSite.
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
```

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.

4. If you used metrics with the old CentraSite and want to migrate the old XML metrics objects, download the objects using curl to a file in your file system. A sample curl command is shown below.

```
curl -u user_name:password -F _querysearchmode=nonserialized -F _xql=*  
-o metrics_file_name http://host:port/CentraSite/CentraSite/RuntimeMetrics
```

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

5. Shut down the old 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.

## Migrate 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.

### Run the Migration Command

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 Upgrade

1. If you use encrypted entries in the *jaas.config* file or in other properties files, you must enter the value stored there again in plain text.
  - a. Open the *jaas.config* file in the new *Software AG\_directory/profiles/CTP/configuration* folder and search for entries starting with "@secure."
  - b. Re-enter passwords or other information in plain text.
  - c. Open all files in the new *Software AG\_directory/profiles/CTP/configuration/com.softwareag.platform.config.propsloader* folder the names of which start with "com.softwareag.security.ldap.server." and end with ".properties".

- d. Search for entries which start with "@secure." and re-enter passwords or other information in plain text.

After you save the files, the passwords are re-encrypted within a few seconds.

2. Start the new CentraSite.
3. 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.
4. If you used metrics with the old CentraSite and downloaded the old XML metrics objects earlier in this procedure, convert the objects to JSON and load them to CentraSite using the `CentraSiteToolbox` command. A sample `CentraSiteToolbox` command is shown below.

```
CentraSiteToolbox ConvertAndLoadMetrics.jar
-dburl http://host:port/CentraSite/CentraSite
-user user -password password -xmlfile metrics_file_name
```

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.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.

- You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.





# 14 Upgrade MashZone NextGen

---

■ Install the New Product .....	122
■ Upgrade from 10.1 or 10.3: Migrate MashZone NextGen .....	123
■ Upgrade from 9.10 or 9.12: Migrate MashZone NextGen .....	125
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	130

**Note:**

There is no direct path from MashZone 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.

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Upgrade from 10.1 or 10.3: Migrate MashZone NextGen

### Prepare the Old Installation

1. Shut down the old installation.

**If the product is running** Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

2. 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.
3. 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.
4. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level jvm directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.
  - a. On the product's old machine, open a command window or shell.
  - b. Go to the Software AG directory that contains the old product and enter this command:
 

```
jar cfM mzng.zip MashZoneNG
```
  - c. 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.

## Migrate 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.

### Run the Migration Utility

This migration utility runs without prompting for any information. If an error occurs, the utility exits. The utility does not revert the new product installation. You can address the issue and rerun the utility.

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 *migrateold\_releasesbs.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
```

## Complete the 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.

### Prepare the Old Installation

1. 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.

2. Shut down the old installation.

#### If the product is running

As a Windows service	Shut down from the Windows Services window. Services are listed as <i>Software AG product release</i> .
----------------------	---

As a Windows application	Shut down from the Windows start menu. Applications are listed as <b>Software AG &gt; Stop Servers &gt; product</b> .
--------------------------	---

On a UNIX system	Use the shut down instructions in the product documentation for your old release.
------------------	---

## Set Up the MashZone NextGen Repository

### Set Up the 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 the 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

Database	JDBC Driver Jar File
SQL Server	jtids-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\_Software**AG\_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 the Old Repository to the 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
```

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 15 Upgrade from 9.9: Upgrade Mobile Designer

---

■ Install the New Product .....	132
■ Prepare the Old Installation .....	132
■ Migrate Mobile Designer .....	133
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	133

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Installation

---

Shut down the old installation.

### If the product is running... Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

## 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.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.

- Specify the correct host names when you configure the new product and when you set up connections to other products.
- 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 16 Upgrade webMethods OneData

---

■ Gather Information for Installation .....	136
■ Install the New Product .....	136
■ Prepare the Old Installation .....	137
■ Migrate Database Components .....	138
■ Migrate webMethods OneData .....	138
■ Complete the Upgrade .....	139
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	140

## Gather Information for Installation

---

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).

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

On the product installer panel or in response to a prompt, enter the values you noted earlier. Make sure the database connections point to the webMethods OneData database components you want to use with the new webMethods OneData.

**Important:**

The installer displays an Administrator Password panel for products that use a default password. webMethods OneData does not use this panel, but you must enter a password in order to continue with installation. webMethods OneData will ignore the inputs from this panel.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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, and that Web service clients of the old webMethods OneData will not have to change URLs to access the new webMethods OneData.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be *MIG*, or *UPG*, for example.



- Install the latest migration framework fix. Fix names for the migration framework follow the convention `MIG_release_MigrationFramework_Fixnumber` 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.

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

## Prepare the Old Installation

1. In the old webMethods 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 webMethods 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 webMethods 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.

<i>old_Software AG_directory/</i>	<b>Files to Back Up</b>
<code>profiles/ODE/configuration/tomcat/conf</code>	<code>server.xml</code>
<code>profiles/ODE/configuration</code>	<code>custom_wrapper.conf</code> and <code>jaas.config</code>
<code>profiles/ODE/workspace/webapps/onedata/WEB-INF</code>	<code>web.xml</code>
<code>profiles/ODE/configuration/com.softwareag. platform.config.propsloader</code>	All <code>com.softwareag.jaas. realm.pid-SSO_realm_name</code>

4. Shut down the old installation.

**If the product is running**

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

**Note:**

It is especially important to shut down all old webMethods OneData instances that connect to database components you are going to migrate.

5. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level jvm directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.
  - a. On the product's old machine, open a command window or shell.
  - b. Go to the Software AG directory that contains the old product and enter this command:
 

```
jar cfM ZIP_file_name.zip install/products profiles/ODE/bin/onedata/config/*
```
  - c. 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.

## Migrate Database Components

---

For instructions on migrating database components for webMethods OneData, see [“Migrate Database Components” on page 153](#).

## Migrate webMethods OneData

---

### Configurations, Data, and Assets that Will be Migrated

- Configuration data except for Log4j properties.

## Run the Migration Utility

This migration utility runs without prompting for any information. If an error occurs, the utility exits. The utility does not revert the new product installation. You can address the issue and rerun the utility.

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*.

On the machine that hosts the new webMethods 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
```

## Complete the 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 webMethods 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 webMethods 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 webMethods 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
```

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.
  - 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.



# 17 Upgrade Zementis Predictive Analytics

---

■ Install the New Product .....	144
■ Prepare the Old Installation .....	144
■ Migrate Zementis Predictive Analytics .....	145
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	145

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Installation

---

1. Go to the *old\_Software AG\_directory/Zementis/adapa-app/migration* directory. Back up Zementis Predictive Analytics models and resources by running the command below. 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.

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

2. Shut down the old installation.



**If the product is running Take this action...**

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

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

## Migrate Zementis Predictive Analytics

### Assets that Will be Migrated

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

### Run the Migration Command

1. Go to the `new_Software AG_directory/Zementis/adapa-app/migration` directory.
2. 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-yyyyymmdd-hhmmss_directory
```

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.

- 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.

# 18 Upgrade Digital Event Services (Deprecated)

---

■ Digital Event Services Deprecation .....	148
■ Install the New Product .....	148
■ Prepare the Old Installation .....	148
■ Migrate Digital Event Services (Deprecated) .....	149
■ Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup .....	150

## Digital Event Services Deprecation

---

Software AG Digital Event Services is deprecated. For more information about possible replacement products, contact Software AG Global Support.

## Install the New Product

---

Follow the instructions in *Using Software AG Installer, Installing Software AG Products*, and in this section to install your new product.

**Note:**

When you install new products on the same machine as old products, the installer often allows you to assign ports used by old products to new products as well, even if the old products are 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.

Install the latest updates on your new product from the Software AG Installer. Product fix names typically follow the convention *product [subcomponent] release Fix number [platform]*. In addition, install the applicable fixes below.

- For some products, fixes relating to migration are separate from the product fixes. Install the latest of these migration fixes on all new products. For migration fixes, the *subcomponent* part of the product fix name might be MIG, or UPG, for example.
- Install the latest migration framework fix. Fix names for the migration framework follow the convention *MIG\_release\_MigrationFramework\_Fixnumber* 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.

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

## Prepare the Old Installation

---

1. Shut down the old installation.

**If the product is running** Take this action...

As a Windows service Shut down from the Windows Services window. Services are listed as Software AG *product release*.

As a Windows application Shut down from the Windows start menu. Applications are listed as **Software AG > Stop Servers > product**.

On a UNIX system Use the shut down instructions in the product documentation for your old release.

2. If your old and new installations are on different machines, create a ZIP file of the old product installation to use as the migration source. The instructions in this section 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. On some systems, the lower-level *jvm* directory name includes additional information, such as */jvm/jvm160\_32*, or */jvm/jvm170*, or */jvm/jvm\_64*.

- a. You can reduce the ZIP file size by first moving the log files out of the *old\_Software AG\_directory/profiles/CTP/logs* directory.
- b. On the product's old machine, open a command window or shell.
- c. If your installation includes the file `migrate.{bat|sh}` in the *new\_Software AG\_directory/common/migrate/DigitalEventServices/bin* directory, run this command to create the ZIP file:

```
jar cfm ZIP_file_name.zip install/products
jar ufm ZIP_file_name.zip common/DigitalEventServices
```

- d. If the following directories exist, add them to the ZIP file:

```
jar ufm ZIP_file_name.zip profiles/product
/configuration/DigitalEventServices
```

- e. 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.

## Migrate Digital Event Services (Deprecated)

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.

## 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.

## Run the Migration Utility

This migration utility runs without prompting for any information. If an error occurs, the utility exits. The utility does not revert the new product installation. You can address the issue and rerun the utility.

Make sure none of the new products are running.

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 *migrateold\_releasesbs.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
```

## Complete the 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.

## Configure Products, Update Host Names, Adjust Assets for Release Changes, Re-enable Auto-Startup

---

Perform the applicable tasks below.

- Configure your new product using instructions in the product documentation for the new release.
- If you installed the new product on a different machine than the old product:
  - Update host names in the new product and in the product database tables.
  - Specify the correct host names when you configure the new product and when you set up connections to other products.

- 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 readme information from your old product release+1 up to and including the new product release. For example, if your old product release is 9.9, read the product readme information for product releases 9.10 through 10.11. Also read the readme information for the same releases of Software AG Infrastructure. Product readmes contain this information:
  - Critical information and known and resolved issues for your product.
  - 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 product changes.

Readmes are available on the Software AG Documentation website.

- After installation, you might have set Windows services for the product 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 product and when you want to start running the new one, 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.
  - If you installed the new product on a new machine, you can reclaim disk space on the new machine by deleting the ZIP file containing the old installation and any files extracted from the ZIP file.
  - You can reclaim space on the old machine by uninstalling the old product using the Software AG Uninstaller for the old release, or Command Central.





# 19 Migrate Database Components

---

■ Requirements and Cautions .....	154
■ Prepare Database Components for Migration .....	154
■ Migrate Database Components Using Database Component Configurator .....	155

## Requirements and Cautions

---

This section lists important considerations for migrating database components:

- Migrate database components after you install the new products and prepare the old installations.
- Use the same type of RDBMS in the new environment that you used in the old environment. The use of a different RDBMS in the new environment is not supported, therefore, cross RDBMS migrations are also not supported.
- The new release might require changes to the database components, such as new tables, columns, keys, or indexes. In [“Migrate Database Components Using Database Component Configurator” on page 155](#), 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

---

- 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, webMethods 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 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} --action migrate --dbms
{db2luw|mysql|oracle|sqlserver|pgsql|tibero}
{--component db_component[,db_component...]} | --product product[,product...]}
--version latest --url db_server_URL --user existing_db_user --password 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.

### Important:

Migrate the Integration Server or Microservices Runtime database components before migrating other database components.

The table below shows the values to specify for the `--component` or `--product` parameter.

If you are upgrading...	Specify this value...
ActiveTransfer Server	<code>--component {ActiveTransfer ActiveTransferArchive}</code>
API Gateway	<code>--component APISGatewayEvents</code>
Archive database component	<p>This database component is used to archive ISCoreAudit and ProcessAudit data.</p> <pre data-bbox="440 590 1370 632">--component Archive</pre>
Integration Server	<pre data-bbox="440 653 1370 716">--component {ISInternal ISCoreAudit CrossReference DocumentHistory DistributedLocking}</pre> <p data-bbox="440 747 1370 810">If the listed database components are in the same schema, you can specify the following instead:</p> <pre data-bbox="440 831 1370 863">-product IS</pre>
Microservices Runtime	<pre data-bbox="440 898 1370 961">--component {ISInternal ISCoreAudit CrossReference DocumentHistory DistributedLocking}</pre> <p data-bbox="440 993 1370 1056">If the listed database components are in the same schema, you can specify the following instead:</p> <pre data-bbox="440 1077 1370 1108">-product MSR</pre>
My webMethods Server	<pre data-bbox="440 1129 1370 1161">--component {MywebMethodsServer}</pre> <p data-bbox="440 1192 1370 1224">Or:</p> <pre data-bbox="440 1245 1370 1276">-product MWS</pre> <p data-bbox="440 1308 1370 1381"><b>Note:</b> The CentralConfiguration database component is automatically migrated.</p>
webMethods OneData	<code>--component {webMethodsOneDataMetadata webMethodsOneDataReleaseArea webMethodsOneDataWorkArea}</code>
Optimize	<pre data-bbox="440 1493 1370 1556">--component {Analysis ProcessTracker ProcessAudit DataPurge DatabaseManagement}</pre> <p data-bbox="440 1587 1370 1650">If the listed database components are in the same schema, you can specify the following instead:</p> <pre data-bbox="440 1671 1370 1703">-product OPTI</pre> <p data-bbox="440 1734 1370 1850"><b>Note:</b> The OperationManagement database component is automatically migrated.</p>

If you are upgrading...	Specify this value...
Process Engine	<p>These database components contain business process run time data for in-progress, completed, or failed business process instances. Migrate them 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.</p>
	<pre>--component {ProcessEngine ProcessAudit}</pre>
	<p>If the listed database components are in the same schema, you can specify the following instead:</p>
	<pre>-product PRE</pre>
Rules Engine	<pre>--component BusinessRules</pre>
Trading Networks	<pre>--component {TradingNetworks TradingNetworksArchive}</pre>
	<p>If the listed database components are in the same schema, you can specify the following instead:</p>
	<pre>-product TN</pre>



# 20 Upgrade Using Command Central

---

■ Supported Upgrade Scenarios .....	160
■ Upgrade Approaches .....	161
■ Upgrade Products in a Single Environment .....	161
■ Upgrade Products Using Software Stacks (10.1 and Later) .....	163

## Supported Upgrade Scenarios

---

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

### 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.

## Upgrade Approaches

- If you want to upgrade products in a single environment using Command Central, follow the instructions in this chapter. 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 the other chapters in this guide if you cannot script them.
- If you want to simplify upgrade now and in the future, you can use Command Central software stacks, 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 chapter.

## Upgrade Products in a Single Environment

Read all chapters in this guide to understand all information and tasks beyond Command Central, such as cluster upgrade guidelines, preparing the old installation, and manual tasks that need to be performed after data migration.

Follow the instructions in *Software AG Command Central Help* to install your new products and fixes, install fixes on your old products, shut down the old products, and create ZIP files of the old installations to use as sources for cross-host migration.

The preparation steps for the old Integration Server or Microservices Runtime include suspending and draining triggers. You can perform this task from 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*. To check the progress of the migration in Command Central, click **Jobs** on the Command Central web user interface.

```
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 from 9.9 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"
```

Use the commands below to migrate the product configurations, data, and assets.

Product	Migration Command
Application Platform	<pre>sagcc exec administration product target_node_alias PLSCore migration migrate {srcDir srcFile}=full_path_to_{old_Software AG_directory ZIP file}</pre>
Digital Event Services	<pre>sagcc exec administration product target_node_alias DEVRuntime migration migrate {srcDir srcFile}=full_path_to_{old_Software AG_directory ZIP file}</pre>
Integration Server Runtime and Hosted Wm Packages Command	<p>If you want to run a custom migration, first run the migration utility manually, and then specify the resulting migrate.dat file on the importFile parameter.</p> <p>If you are upgrade on the same machine and you want to migrate all instances, do not specify the instanceName parameter. If you are upgrade Integration Server on a new machine, migrate one instance at a time to avoid causing concurrency issues with your database.</p> <pre>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]</pre>
My webMethods Server	<p>If you want to run a custom migration, first run the migration utility manually, and then specify the resulting migrate.dat file on the importFile parameter.</p> <p>If you want to migrate all instances, do not specify the instanceName parameter.</p> <pre>sagcc exec administration product target_node_alias</pre>

Product	Migration Command
	<pre data-bbox="488 247 1453 422">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]</pre> <p data-bbox="488 464 1453 737"><b>Note:</b> 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.</p>
Universal Messaging	<p data-bbox="483 772 1474 877">If you want to run a custom migration, first run the migration utility manually, and then specify the resulting migrate.dat file on the <code>importFile</code> parameter. If you want to migrate all instances, do not specify the <code>instanceName</code> parameter.</p> <pre data-bbox="488 894 1453 1035">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...]]</pre>

## Upgrade Products Using 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.

For more information, go to <https://tech.forums.softwareag.com/t/announcing-one-click-upgrades-for-webmethods/242399> on the Software AG TECHcommunity website. You can find videos on upgrading using stacks on Youtube; search on "Software AG Command Central" to find videos from which to choose.

1. Complete the tasks in [“Prepare for Upgrade” on page 11](#) and the *Prepare the Old Installation* sections in the product-specific chapters 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. 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.

For instructions on installing fixes, see *Software AG Command Central Help*.

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, 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, the table below explains the next action to take.

If your Windows release...	Take this action...
Has out-of-the-box SSH support	Use the default SSH server on the Windows machine
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 <a href="https://cygwin.com/install.html">https://cygwin.com/install.html</a> or follow the instructions in the <i>Using Cygwin to Configure OpenSSH When Installing Platform Manager on a Remote Windows Machine</i> 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. For instructions, see *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. Upgrade the stack as described below.
  - a. Click **Stacks** at the top of the GUI.
  - b. On the stack to upgrade, click , click **Upgrade stack**, click the type of upgrade to perform, and provide the requested values.
  - c. Click **Dry run** at the bottom of the configuration page. Command Central generates a composite template for each runtime component in the stack.
  - d. You might want or need to edit the generated templates for reasons such as the following:
    - To reflect changes, deprecations, and removals specified in the product readmes.
    - Command Central indicates that a product in a template does not exist in the repositories for the new release, so you need to remove the product entry from the template or specify the replacement product.
    - Upgrade from 10.1 or 10.3: The stack contains a database layer, which means you need to copy the database product IDs from the database scripts section to the products section.

To edit the generated templates:

- a. Click **Templates** at the top of the GUI. You will see a list of templates named `migration_generated_stack_name`.
- b. Download a template and edit with a text editing tool.
- c. Click  to import the template.
- d. Repeat the steps above for other templates as necessary.

- e. Click **Stacks** again, click , click **Upgrade stack** again, and choose to use existing templates.
  - e. When the generated templates are ready, you can perform the upgrade by clicking **Upgrade stack** at the bottom of the configuration page, or you can use the templates in automated upgrades.
10. Complete the product upgrades as explained in the product-specific chapters in this guide.