

Upgrading Software AG Products

Version 9.10

April 2016

This document applies to webMethods Product Suite Version 9.10 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 © 2011-2016 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 <http://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 <http://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 <http://softwareag.com/licenses> and/or in the root installation directory of the licensed product(s).

Table of Contents

About this Guide.....	9
Document Conventions.....	9
Online Information.....	10
Upgrades Covered in This Guide.....	11
Upgrades Covered in This Guide.....	12
Upgrades Not Covered in This Guide.....	12
Release Numbering Exceptions.....	13
Documentation Needed to Perform the Upgrade.....	13
Critical Factors and Requirements for Successful Upgrade.....	15
Preparation and Planning.....	16
Testing and Troubleshooting.....	17
General Upgrade Procedure Requirements.....	18
Command Central Requirements.....	19
API-Portal Requirements.....	21
CentraSite Requirements.....	21
Integration Server Clustering Requirements.....	21
Mobile Administrator Requirements.....	22
My webMethods Server Requirements.....	22
Environment Variable.....	22
My webMethods Server Clustering Requirements.....	22
Database Requirements, Recommendations, and Preparation.....	23
Install New Products and Begin Migration.....	25
Read and Complete All Previous Chapters.....	26
Before Installing webMethods Broker.....	26
Gather Values for OneData Installation.....	26
Install the New Products.....	27
Hardware Requirements.....	27
Run the Software AG Installer.....	27
Complete Post-Installation Tasks.....	29
Complete Installation of the New Products.....	29
Install Latest Fixes on New Products.....	29
Disable Windows Services.....	29
Prepare the Old Environment for Upgrade.....	30
Install Fixes on Old Products.....	30
Prepare the Old API-Portal.....	30
Prepare the Old CentraSite.....	31
Prepare the Old Software AG Designer.....	31
Export Integration Server Definitions.....	31
Export CloudStreams Server Definitions and Projects.....	31

Export Event Bus Console Configurations.....	32
Export Preferences.....	32
Prepare the Old Integration Server.....	32
Start and Connect Products.....	32
9.0 Upgrade: Suspend Document Retrieval and Make Sure All Messages Have Been Sent.....	33
9.5 Upgrade: Suspend Triggers and Make Sure Queues are Drained.....	33
9.6, 9.7, 9.8, 9.9 Upgrade: Suspend Triggers and Make Sure Queues are Drained.....	33
Make Sure Queues are Drained and Complete or Suspend Business Processes.....	34
Prepare the Old My webMethods Server.....	34
Prepare the Old OneData.....	34
Prepare the Old Optimize.....	35
Shut Down the Old Products.....	35
Migrate Database Components.....	36
Shut Down Products that Connect to Database Components.....	36
Before You Migrate Trading Networks Database Components.....	36
Special Considerations for Migrating Other Database Components.....	37
Install Fixes on Database Migration Scripts.....	37
Migrate Database Components.....	38
Prepare to Migrate Product Assets.....	39
Migration Utilities.....	39
Create Zip Files as Source of Assets to Migrate.....	40
Create a Zip File for My webMethods Server.....	40
Create Zip Files for All Other Products.....	40
Migrate the Software AG Infrastructure.....	42
Before Migrating.....	42
Run the Software AG Infrastructure Migration Utility.....	42
Migrate LDAP Server Configuration.....	43
Update the Software AG Runtime NERV Configuration.....	43
Migrate Java Service Wrapper Comments and #include Directives.....	43
Where to Go from Here.....	44
Migrate Terracotta, Universal Messaging, and webMethods Broker.....	45
Read and Complete All Previous Chapters.....	46
Migrate Terracotta.....	46
Migrate Universal Messaging.....	46
Before Migrating.....	46
Run the Universal Messaging Migration Utility.....	46
Perform a Custom Migration.....	47
Migrate Using Custom Imported Settings with Prompting.....	47
9.0 through 9.7 Upgrade: Migrate Using Custom Imported Settings without Prompting (Silent).....	47
9.8 and 9.9 Upgrade: Migrate without Using Custom Imported Settings or Prompting (Silent).....	48
Migrate Custom Changes and Data Directory.....	48

9.0, 9.5, 9.6 Upgrade: Prepare to Use webMethods Messaging (Native Pub-Sub).....	49
Migrate webMethods Broker.....	49
Types of Migration.....	49
Migrate from Old webMethods Broker to New Universal Messaging.....	49
Migrate from Old webMethods Broker to New webMethods Broker.....	49
Create a Broker Server and Use the Old Storage.....	50
Migrate My webMethods Server.....	53
Read and Complete All Previous Chapters.....	54
Before Migrating.....	54
Run the My webMethods Server Migration Utility.....	54
Perform a Custom Migration.....	54
Migrate Using Imported Settings.....	55
Migrate Using Custom Imported Settings with Prompting.....	55
Migrate Using Default Imported Settings and the Live Database without Prompting (Silent).....	55
Migrate Using Custom Imported Settings without Prompting (Silent).....	56
Migrate Java Server Wrapper Comments and #include Directives.....	56
Initialize a My webMethods Server Instance.....	56
Switch from Clone to Live Database.....	57
Migrate Integration Server and Hosted Wm Packages.....	59
Read and Complete All Previous Chapters.....	60
Packages.....	60
Before Migrating.....	60
Run the Integration Server Migration Utility.....	61
Perform a Custom Migration.....	61
Migrate Using Imported Settings.....	63
Migrate Using Custom or Default Imported Settings with Prompting.....	64
Migrate Using Custom or Default Imported Settings without Prompting (Silent).....	64
Other Actions Performed by the Migration Utility.....	64
Complete the Integration Server Upgrade.....	65
Migrate Java Server Wrapper Customizations.....	65
Customize Integration Server Startup Files.....	65
Update WSDLs.....	65
Synchronize and Update Serial webMethods Messaging Triggers.....	66
Connect Integration Server Functions to Database Components.....	66
Update Host Names.....	66
Update the Integration Server NERV Configuration.....	67
9.0 Upgrade: Migrate Custom Jar Files.....	67
Complete the ActiveTransfer Upgrade.....	67
Complete the CloudStreams Upgrade.....	68
Complete the Trading Networks Upgrade.....	69
Complete Adapter Upgrades.....	69
webMethods Adapter for JDBC.....	70
webMethods Package for Microsoft .NET.....	70

webMethods PeopleSoft EnterpriseOne Adapter.....	70
webMethods SAP Adapter.....	70
Complete eStandards Module Upgrades.....	71
webMethods SWIFTNet Module.....	72
Migrate Software AG Designer and Business Process Runtime.....	73
Read and Complete All Previous Chapters.....	74
Migrate Software AG Designer.....	74
Before Migrating.....	74
Migrate Apama, Application Platform, Business Process, and Business Rule Projects.....	75
Migrate Task Application Projects.....	76
Redo Attachment List Control Customizations in CAF Applications.....	76
Migrate CloudStreams Server Definitions and Projects.....	76
Migrate Integration Server Definitions.....	77
Migrate the Business Process Runtime.....	77
Migrate Closed Loop Analytics.....	78
Migrate Optimize and Infrastructure Data Collector.....	81
Read and Complete All Previous Chapters.....	82
Configure and Start the Terracotta Server Array.....	82
Update Connection to Other Products.....	82
Update Connection to Process Engine.....	82
Update Connection to Optimize Support Package.....	82
Migrate Infrastructure Data Collector.....	83
Run the Infrastructure Data Collector Migration Utility.....	83
Before Migrating.....	83
Perform a Custom Migration.....	83
Migrate Using Imported Settings.....	84
Migrate Using Custom or Default Imported Settings with Prompting.....	84
Migrate Using Custom or Default Imported Settings without Prompting.....	85
Migrate Java Server Wrapper Comments and #include Directives.....	85
9.7 Upgrade: Start Monitoring 4.1 or Later Terracotta Server Arrays.....	85
Deploy the Optimize Environment.....	85
Not Reuse the Old Optimize CCS Environment Definition.....	85
Reuse the Old Optimize CCS Environment Definition.....	86
Deploy the New Optimize Environment.....	87
Switch from webMethods Broker to Universal Messaging.....	87
Migrate the Asset Build Environment, Deployer, and Application Platform.....	89
Read and Complete All Previous Chapters.....	90
Migrate the Asset Build Environment.....	90
Migrate Deployer.....	90
Before Migrating Deployer Settings, Server Aliases, Target Groups, and Projects.....	90
Migrate All Deployer Settings, Server Aliases, Target Groups, and Projects Using the Integration Server Migration Utility.....	91

Migrate Deployer Settings, Server Aliases, Target Groups, and Projects Using the Deployer Migration Feature.....	91
Start the Old and New Deployers and Migrate Global (Default) Settings.....	91
Migrate Server Aliases (Connection Information).....	92
Migrate Target Groups.....	92
Migrate Deployment Projects.....	93
Migrate Application Platform.....	93
Migrate Apama.....	95
Read and Complete All Previous Chapters.....	96
Before Migrating.....	96
Complete the Apama Upgrade.....	96
Migrate API-Portal and ContraSite.....	97
Read and Complete All Previous Chapters.....	98
Migrate API-Portal.....	98
Migrate ContraSite.....	99
Before Migrating.....	99
Migrate the Data.....	100
Complete the Upgrade.....	101
Migrate MashZone NextGen.....	103
Read and Complete All Previous Chapters.....	104
Overview.....	104
Set Up the MashZone NextGen Repository.....	104
Set Up Repository When Using the Embedded Derby Database.....	104
Set Up Repository When Using Any Other Database.....	105
Update the MashZone NextGen Configuration and Copy Extensions.....	105
Update When an External Configuration Directory Was Used.....	106
Update When No External Configuration Directory Was Used.....	106
Copy Integrated MashZone Data and Run Upgrade Commands.....	106
3.7 or 3.8 Upgrade: Copy Integrated MashZone Data and Run Upgrade Commands....	106
3.9 Upgrade: Copy Integrated MashZone Data and Run Upgrade Commands.....	107
9.9 Upgrade: Upgrade JDBC Configuration and Run Upgrade Commands.....	108
3.6 or 3.7 Upgrade: Update Mashups that Use RAQL Queries.....	108
Update the Snapshots Repository Connection Information.....	109
Migrate the Mobile Suite.....	111
Read and Complete All Previous Chapters.....	112
9.7, 9.8, or 9.9 Upgrade: Migrate Mobile Designer.....	112
Migrate Mobile Administrator.....	112
Overview.....	112
Run the Updater and Verify Configurations.....	113
Update Build Nodes.....	113
Update the App Store Client.....	113
Migrate OneData.....	115

Read and Complete All Previous Chapters.....	116
Before Migrating.....	116
Run the OneData Migration Utility.....	116
Complete the OneData Upgrade.....	116
Migrate Software AG Event Routing and Perform Final Upgrade Tasks for All Products.....	119
Migrate Software AG Event Routing.....	120
Before Migrating.....	120
Run the Software AG Event Routing Migration Utility.....	120
Migrate Undelivered Events.....	120
9.9 Upgrade: Migrate Event Persistence Services.....	121
Update Paths.....	121
Complete Final Upgrade Tasks for All Products.....	121

About this Guide

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

Note: The information provided in this guide for Terracotta and Universal Messaging applies to when you are using those products with webMethods products. For information on those products in other contexts, see the product documentation.

Document Conventions

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

Convention	Description
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information

Software AG Documentation Website

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

Software AG Empower Product Support Website

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

To submit feature/enhancement requests, get information about product availability, and download products, go to [Products](#).

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

Software AG TECHcommunity

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

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

1 Upgrades Covered in This Guide

■ Upgrades Covered in This Guide	12
■ Upgrades Not Covered in This Guide	12
■ Release Numbering Exceptions	13
■ Documentation Needed to Perform the Upgrade	13

Upgrades Covered in This Guide

This guide explains how to upgrade Apama, Terracotta, and webMethods products from release 9.0, 9.5, 9.6, 9.7, 9.8, or 9.9 to release 9.10 using a side-by-side procedure.

The information provided in this guide for Terracotta and Universal Messaging applies to when you are using those products with webMethods products. For information on those products in other contexts, see the product documentation.

Note: Software AG tests migration across machines that have the same hardware and operating systems. Integration Server also tests migration across machines that have different hardware or different operating systems. However, for other products, migration across machines that have different hardware or operating systems might have unpredictable results, especially when one machine is Windows and the other is non-Windows.

Upgrades Not Covered in This Guide

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

Upgrade	Information
From a release prior to 9.0	See <i>Supported Upgrade Paths</i>
Apama releases prior to 9.9	See the Apama <i>What's New</i> guide for your pre-9.9 release
ApplnX	See ApplnX documentation
webMethods Broker to Universal Messaging	See <i>Migrating from webMethods Broker to webMethods Universal Messaging</i>
Content Service Platform	Contact Software AG Global Consulting Services
MashZone or Process Performance Manager	See MashZone or Process Performance Manager documentation
Mobile Administrator on a virtual machine (this guide explains how	Uninstall Mobile Administrator from the virtual machine (see the installation guide for the old release) and install the

Upgrade	Information
to upgrade Mobile Administrator on a physical machine)	new Mobile Administrator (see <i>Installing Software AG Products</i> <i>new_release</i>)
Presto (now MashZone NextGen) releases prior to 3.6	Contact Software AG Global Consulting Services
Terracotta when not used with webMethods products	See Terracotta BigMemory Max documentation
Universal Messaging when not used with webMethods products	See Universal Messaging documentation

Release Numbering Exceptions

The products below did not follow general release numbering.

Product	Release Numbering
ActiveTransfer	In the webMethods 9.9 and 9.10 releases, the ActiveTransfer release number is still 9.8 and is the same release that was available with the webMethods 9.8 release.
webMethods Broker	In the webMethods 9.7 and later releases, the webMethods Broker release number is still 9.6 and is the same release that was available with the webMethods 9.6 release.
Presto (now MashZone NextGen)	In the webMethods 9.6, 9.7, and 9.8 releases, the Presto release numbers were 3.7, 3.8, and 3.9, respectively. The Presto release before Presto became part of webMethods was 3.6.
Terracotta	In the webMethods 9.0, 9.5, 9.6, 9.7, 9.8, 9.9 and 9.10 releases, the Terracotta release numbers were 3.7.4, 3.7.6, 3.7.6, 4.1.4, 4.3, 4.3.1, and 4.3.2 respectively.

Documentation Needed to Perform the Upgrade

You will need the documentation listed below to perform the upgrade.

- Release notes for the old release + 1 through the new release; these list new features

- Product readmes for the new release; these describe changes to product behavior; deprecations, removals, and additions of services, APIs, and other items; critical information about the new product release; and known and resolved issues
- Fix readmes for fixes you install on new products; these list resolved issues and changes in behavior
- *System Requirements for Software AG Products*
- *Supported Upgrade Paths for Software AG Products*
- *Using the Software AG Installer*
- *Installing Software AG Products*
- *Using the Software AG Update Manager* (latest release)

You might need the documentation listed below to perform this upgrade, depending on the products you are upgrading. The release of the documentation you need is the new release.

- *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*
- *Administering webMethods OneData*
- *Administering webMethods Optimize and Configuring BAM*
- MashZone NextGen documentation
- *Using Terracotta with webMethods Products and Terracotta BigMemory Max documentation*

2 Critical Factors and Requirements for Successful Upgrade

■ Preparation and Planning	16
■ Testing and Troubleshooting	17
■ General Upgrade Procedure Requirements	18
■ Command Central Requirements	19
■ API-Portal Requirements	21
■ CentraSite Requirements	21
■ Integration Server Clustering Requirements	21
■ Mobile Administrator Requirements	22
■ My webMethods Server Requirements	22
■ Database Requirements, Recommendations, and Preparation	23

Preparation and Planning

Software AG strongly recommends that you plan and prepare for your upgrade before performing the procedure described in this guide. Planning and preparing can include the tasks listed below.

- Read this guide in its entirety so you are familiar with all tasks you will need to perform.
- Read the product release notes. The release notes provide information on new functionality. Read the information for every release for your old release + 1 through the new release.
- Read the product readmes for the new release, including the readme for the Software AG Infrastructure. All readmes are available on the Software AG Documentation website. The product readmes contain this information:
 - Critical information and known and resolved issues for your products.
 - Changes (that is, additions, changes, deprecations, and removals) to product behavior, services, parameters, properties, and APIs. This information is especially important because you might need to modify product files or assets after migration to accommodate the changes.

Product readmes contain information for the 9.5 release and later. Read the information for your old release + 1 through the new release. For example, if you are upgrading from 9.0 to 9.9, read the information for releases 9.5 through 9.9.

- Define an upgrade approach and a cut-over plan. If you have a large landscape, you might want to do a phased upgrade.
- Set up operating guidelines that cover normal project management requirements across all affected groups in your organization.
- Define an approach to source control, and set up release management tools and procedures that support the migration of data into target environments.
- Account for externals to your Software AG environment, such as clients that communicate with your and Software AG products and that might need endpoint URL, host name, or IP address updates.
- As part of upgrading, you will take a baseline of the old environment and use that baseline to perform the upgrade procedure. Software AG recommends a code freeze while you are migrating the baseline to the new environment.
- Many upgrade issues are caused by forgotten environmental settings. Plan, document, and test these thoroughly.
- For complex projects that require extensive system testing, extended code freeze might not be possible. In this case, define a change management strategy to track code changes that occur in the old environment. After testing is complete on the

baseline, this will enable you to merge the code changes into the baseline and re-test the new baseline.

If you need help creating your planning approach, contact Software AG Global Consulting Services. You can also go to the [Software AG TECHcommunity](#) and find tips and tricks and other information in the upgrade wiki (Wiki > webMethods > Upgrade).

Testing and Troubleshooting

Testing can include the tasks listed below.

- Software AG strongly recommends that you upgrade in a controlled test environment and test that environment for proper operation before promoting to your production environment.
- Provision new environments to support the upgrade. Environments in which you conduct testing should mirror the production environment.
- Obtain test data from the source environment and establish baseline test results.
- Many upgrade issues are caused by forgotten environmental settings. Plan, document, and test these thoroughly.
- Identify test tools for unit, functional, and non-functional regression testing.
- Define acceptance and sign off criteria.
- Automate a core set of regression tests for the upgrade. Software AG recommends focusing your tests most heavily on product areas that have undergone the most change. The readmes and release notes provide all product change information (see "[Preparation and Planning](#)" on page 16 for details).
- If upgrading to new machines, make sure connections and firewalls are open to back-end systems before starting any system testing.
- Use an issue management system to track defects and issues.
- If code freeze of the old environment during test is not possible, merge code changes from the old environment into the baseline and re-test the new baseline.

If you need help creating your testing approach, contact Software AG Global Consulting Services.

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

- Look in log files.

Type of Log	Location
Installation and uninstallation	<i>new_Software AG_directory</i> \install\logs directory

Type of Log	Location
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 search using keywords such as upgrade.
- Go to the [Software AG TECHcommunity](#) and:
 - Join the upgrade discussion forum (Forums > webMethods > Upgrade).
 - Read and share tips and tricks and other information in the upgrade wiki (Wiki > webMethods > Upgrade).

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

General Upgrade Procedure Requirements

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

- Perform the tasks in 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. Make sure to complete the tasks in the following chapters, in order:
 - This chapter.
 - [Install New Products and Begin Migration](#).
 - All chapters for your products.
 - [Migrate Software AG Event Routing and Perform Final Upgrade Tasks for All Products](#).
- After you install the new products, you install the latest fixes (see "[Disable Windows Services](#)" on page 29). 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.
- 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.
- This guide assumes you are using the same RDBMS in the new environment that you used in the old environment. If you want to use a different RDBMS in the new environment, there are special requirements that must be met before you can upgrade your products. Contact Software AG Global Consulting Services for more information.

Command Central Requirements

You can use Command Central to install most products in the new release, and to install fixes on most old and new products. You can also use Command Central to migrate product files and assets for some products from some old releases to the new release. For the list of products that Command Central can install and migrate, and the upgrade paths that Command Central supports for those products, see the Command Central readme.

This section explains how to upgrade Command Central and how to install and upgrade products using Command Central. The section is intended for use with the rest of this guide. You must perform all tasks in this guide for your products, in the order they are written, unless this section indicates otherwise.

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

1. Upgrade Command Central.
 - a. Install Command Central using the instructions in *Installing Software AG Products*.
 - b. If you installed the new Command Central on a different machine than the old installation, create a Zip file of the old Command Central. For instructions, see "[Create Zip Files for All Other Products](#)" on page 40.
 - c. Use the Command Central bootstrapper to migrate configuration files and assets from the old Command Central to the new Command Central. On the bootstrapper command, specify the -m option and set the option to the Software AG directory that contains the old Command Central installation or the Zip file you made in the previous step. For a list of files and assets that are migrated, see *Software AG Command Central Help*.
 - d. Go to the *Software AG_directory* \profiles\CCE\configuration \com.softwareag.platform.config.propsloader directory in the old and new installations and open the com.softwareag.sso.pid.properties. If a value in the old file is different from the corresponding value in the new file, copy the old value over the new value. In addition, update any absolute file system paths in the new files to point to the new installation. Do not update paths that use

tokens (for example, do not update @path\sag.install.area\common\conf\platform_truststore.jks).

- e. 9.0, 9.5, 9.6, or 9.7 upgrade: If you set a custom password for Platform Manager, set the custom password in the new Command Central.
 - f. Set the LDAP domain property in the new Command Central.
2. Use the new Command Central to set up custom templates that will automate the migration of your products. If you used custom templates to provision your old environment, you will need to make only minor changes to those templates. If you did not use custom templates to provision your old environment, create custom templates in the new Command Central. In the templates, specify the product migration utilities to run, including the Software AG Infrastructure migration utility, and specify the database components to migrate.
 3. This guide explains how to install the new products using Software AG Installer. You can install the new products using Command Central instead, but you must still read and perform the actions described in ["Install the New Products" on page 27](#).
 4. This guide explains how to install fixes on some old products using Software AG Update Manager. You can install the fixes using Command Central instead.
 5. This guide explains how to prepare the old Integration Server for upgrade, including suspending document retrieval and making sure all messages have been sent. When you migrate using Command Central, Command Central automatically quiesces Integration Server to suspend all triggers and drain message queues for you.
 6. This guide explains how to shut down the old products manually. If Command Central managed the old products and the new and old installations are Secure Shell (SSH)-accessible, Command Central shuts down the old products automatically.
 7. This guide explains how to install fixes on the database migration scripts using Software AG Update Manager. You can install the fixes using Command Central instead.
 8. This guide explains how to make Zip files of old installations and copy them to new installations manually when your old and new installations are on different machines. If the new and old installations are SSH-accessible, Command Central automatically creates a Zip file of the entire old Software AG installation directory and copies the Zip file from the old installation to the new installation for you. By default, the Zip file does not include log files and jar files from the old installation.
 9. This guide provides product-specific migration chapters that describe pre- and post-migration tasks you must perform and explain how to run the product migration utilities. Even when you migrate your products using Command Central, you must perform the pre-migration tasks first and the post-migration tasks afterwards.
 10. Perform all tasks in ["Migrate Software AG Event Routing and Perform Final Upgrade Tasks for All Products" on page 119](#).

API-Portal Requirements

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

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

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

CentraSite Requirements

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

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

Integration Server Clustering Requirements

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

The section is intended for use with the rest of this guide. You must perform all tasks in this guide for your products, in the order they are written, with the additional tasks or exceptions noted in this section.

- Use the instructions below with the instructions in ["Install New Products and Begin Migration" on page 25](#).
 - Install a cluster of new Integration Servers that matches your old cluster. You can install the new Integration Servers in parallel.
 - You can install the fixes on the new Integration Servers in parallel.
 - When you shut down the old products, shut down all cluster nodes.
- Use the instructions below with the instructions in ["Migrate Integration Server and Hosted Wm Packages" on page 59](#).

- Run the Integration Server migration utility to migrate server instances from old Integration Server installations to new Integration Server installations. You can migrate instances across Integration Server installations in parallel.
- Make sure the new server instances are processing client requests.
- ["Migrate Software AG Event Routing and Perform Final Upgrade Tasks for All Products" on page 119](#) includes a task to configure your products. For instructions on configuring Integration Server clusters, see the *webMethods Integration Server Clustering Guide*.

Mobile Administrator Requirements

Mobile Administrator 9.10 requires a different license from previous releases. Obtain a Mobile Administrator 9.10 license file from Software AG.

My webMethods Server Requirements

Environment Variable

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

My webMethods Server Clustering Requirements

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

The section is intended for use with the rest of this guide. You must perform all tasks in this guide for your products, in the order they are written, unless this section indicates otherwise.

- Use the instructions below with the instructions in ["Install New Products and Begin Migration" on page 25](#).
 - Install a cluster of new My webMethods Server installations that matches your old cluster. You can install the new My webMethods Servers in parallel.
 - You can install the fixes on the new My webMethods Servers in parallel.
 - You do not need to upgrade the Software AG Infrastructure.

- When shutting down the old products, shut down all cluster nodes.
- Use the instructions below with the instructions in ["Migrate My webMethods Server" on page 53](#).
 - Run the My webMethods Server migration utility to migrate server instances from old My webMethods Server installations to new My webMethods Server installations. You can migrate instances across My webMethods Server installations in parallel.
 - You can initialize the server instances in parallel.
- ["Migrate Software AG Event Routing and Perform Final Upgrade Tasks for All Products" on page 119](#) includes a task to configure your products. For instructions on configuring My webMethods Server clusters, see *Administering My webMethods Server*.

Database Requirements, Recommendations, and Preparation

- Check your RDBMS against *System Requirements for Software AG Products*. If the RDBMS version you are using is not supported by your new products, upgrade to a supported RDBMS version.
- Some products, like Integration Server, Optimize, Process Engine, and Trading Networks, offer features to archive or purge data from their database components. You can reduce the amount of time needed to migrate database components later in this procedure if you archive and purge them now. For instructions, see the product documentation.
- Make a backup of the product databases; shut down all ActiveTransfer, Integration Server, My webMethods Server, OneData, and Optimize instances that connect to database components before making the backup. If you are upgrading My webMethods Server, back up the My webMethods Server installation directory at the same time you back up the database. If you have problems, you will need to restore data from both types of backup.
- 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.

Software AG recommends the following:

- *Installing Software AG Products* describes the basic grants and privileges needed to work with product database components. Before cloning the databases, give the database users that will work with the cloned databases the same basic grants

and privileges that were given to the database users that work with the live databases.

- In some cases, one database user grants permissions to a second user (for example, the process audit database user grants permissions to a second user to archive process audit data). Before cloning the databases, create the second user in the new schema or database.
- Use a separate database user (Oracle) or database (SQL Server or DB2) to host the cloned databases.
- The new release might require changes to the database components, such as new tables, columns, keys, or indexes. You will 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. ["Migrate Database Components" on page 36](#)
- Database migration may take many hours, depending on the volume of data, and is often the critical path for upgrade.

3 Install New Products and Begin Migration

■ Read and Complete All Previous Chapters	26
■ Before Installing webMethods Broker	26
■ Gather Values for OneData Installation	26
■ Install the New Products	27
■ Prepare the Old Environment for Upgrade	30
■ Shut Down the Old Products	35
■ Migrate Database Components	36
■ Prepare to Migrate Product Assets	39
■ Migrate the Software AG Infrastructure	42
■ Where to Go from Here	44

Read and Complete All Previous Chapters

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

Before Installing webMethods Broker

In the webMethods 9.7 and later releases, the webMethods Broker release number is still 9.6 and is the same release that was available with the webMethods 9.6 release. If you have webMethods Broker 9.6 and it resides on its own machine, you can continue using it with your new products. Regardless of whether you install a new webMethods Broker or not, however, you must perform tasks later in this guide for webMethods Broker.

Gather Values for OneData Installation

- Go to the *old_Software AG_directory* \profiles\ODE\configuration \com.softwareag.platform.config.propsloader directory. Open the com.softwareag.catalina.connector.http.pid-*port*.properties file and the com.softwareag.catalina.connector.https.pid-*port*.properties file and note the port numbers.
- Open the file or files below.
 - 9.0, 9.5, 9.6, 9.7, 9.8 upgrade: Go to the *old_Software AG_directory* \profiles\ODE \configuration\tomcat\conf directory and open the context.xml file.
 - 9.9 upgrade: 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* Db1.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 user names for the metadata, work area, and release area schemas. For example, in the metadata schema user name dev2_md, the connection prefix is dev2. Also note any database parameters such as maxIdle, maxActive, and maxWait that are specified for the schemas.

- 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 Products

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

If you are upgrading a cluster of Integration Servers, My webMethods Servers, or Universal Messaging servers, read the clustering requirements for those products in "[Critical Factors and Requirements for Successful Upgrade](#)" on page 15 before proceeding.

Hardware Requirements

Installing Software AG Products contains hardware requirements for each product. However, in some cases you will need additional hard drive space on the target machine.

- If you are upgrading products involving Software AG Designer projects, make sure the target machine has enough space for the projects you will migrate (see "[Migrate Software AG Designer and Business Process Runtime](#)" on page 73).
- If you are upgrading Integration Server, make sure the target machine has enough space for custom packages you will migrate. Custom packages are Integration Server packages created by users in Software AG Designer and business process runtime packages generated by users from Software AG Designer.

Run the Software AG Installer

1. For installation directory, specify a new Software AG installation directory.
2. On the product tree, select the products to install. Also select Database Configuration.
3. For the product panels, follow the instructions in *Installing Software AG Products*, with the additional guidelines below.

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

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

Complete Post-Installation Tasks

Complete Installation of the New Products

Follow the instructions in *Installing Software AG Products* to complete the installation of the new products.

For My webMethods Server, if you installed the new release of on the same machine as the old release, you removed the environment variable WM_HOME from the global path. Do not redefine the environment variable after installing the new My webMethods Server.

Install Latest Fixes on New Products

The instructions in *Installing Software AG Products* include installing the latest fixes on your new products.

- Install the latest *product* fixes on all new products except the infrastructure components named Common Platform and Security Infrastructure. You will apply those fixes after you migrate those products.
- For some products, fixes relating to migration are separate from product fixes. Install the latest *migration* fixes on all new products, including Common Platform and Security Infrastructure. Fix names for these products typically include the letters OSGI and SIN, respectively.
- 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.

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

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

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

Disable Windows Services

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

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

Prepare the Old Environment for Upgrade

Install Fixes on Old Products

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

Prepare the Old API-Portal

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

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

4. Run this command:

```
acc> startall
```

5. Back up the old master tenant to a file.

For	Run these commands...
9.7	<pre>acc> set acc config backup.restore.tenant.app.types=com.aris.umcadmin.y-umcadmin-run-prod acc> set acc config backup.restore.tenant.app.type.names=umc acc> backup tenant master to full_path_to_backup_file username=your_user_name password=your_password</pre>
9.8	<pre>acc> set acc config backup.restore.tenant.app.types=ECP,ADS,UMC acc> backup tenant master to full_path_to_backup_file username=your_user_name password=your_password</pre>
9.9	<pre>acc> set acc config backup.restore.tenant.app.types=ECP,ADS,UMC acc> backup tenant master to full_path_to_backup_file username=your_user_name password=your_password</pre>

6. 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
```

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

For	Run these commands...
9.7	<pre>acc> set acc config backup.restore.tenant.app.types = com.aris.umcadmin.y-umcadmin-run-prod, com.aris.modeling.components.y-server-run-prod, com.aris.adsadmin.y-adsadmin-run-prod, com.aris.runnables.ecp-run-prod acc> set acc config backup.restore.tenant.app.type.names = umc, abs, ads, ecp acc> backup tenant <i>tenant_name</i> to <i>full_path_to_backup_file</i> username=<i>your_user_name</i> password=<i>your_password</i></pre>
9.8	<pre>acc> set acc config backup.restore.tenant.app.types=UMC,ABS,ADS,ECP acc> backup tenant <i>tenant_name</i> to <i>full_path_to_backup_file</i> username=<i>your_user_name</i> password=<i>your_password</i></pre>
9.9	<pre>acc> set acc config backup.restore.tenant.app.types=UMC,ABS,ADS,ECP acc> backup tenant <i>tenant_name</i> to <i>full_path_to_backup_file</i> username=<i>your_user_name</i> password=<i>your_password</i></pre>

8. If you created a customized view of your old API-Portal, back up the view as follows:
 - a. Open API-Portal in a browser and log on with Administrator credentials.
 - b. Go to the **Administration** page. In the **Views** page under the **Customization** section, hover over the name of the custom view and click **Backup**. API-Portal creates a zip file that contains the customized view.
 - c. Save the zip file.

Prepare the Old CentraSite

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

Prepare the Old Software AG Designer

Export Integration Server Definitions

1. In the old Software AG Designer, go to **Window > Preferences**. On the **Preferences** dialog box, in the left navigation bar, go to **Software AG > Integration Servers**.
2. Click **Export** and complete the dialog box. Software AG Designer will save the file with the extension .properties.

Export CloudStreams Server Definitions and Projects

1. To export CloudStreams Server definitions, go to **Window > Preferences**. On the **Preferences** dialog box, in the left navigation bar, go to **Software AG**

- > CloudStreams Servers.** Click **Export** and complete the dialog box. Software AG Designer will save the file with the extension `.properties`.
- Open the CloudStreams Development perspective. In the CloudStreams Governance list, right-click **CloudStreams Governance projects to export**, click **Export**, and complete the dialog box.
- Repeat the previous step to export CloudStreams Provider projects.

Export Event Bus Console Configurations

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

Export Preferences

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

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

Prepare the Old Integration Server

Start and Connect Products

- If you are using webMethods Broker, shut down the new Broker Monitor if it is running. Restart the old Broker Monitor, and then start the old Broker Server.
- Start the old Integration Server and open the old Integration Server Administrator.
- If you are using webMethods Broker, make sure Integration Server is connected to the Broker. If you are using Universal Messaging, make sure Integration Server 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 is connected to the JMS provider.

9.0 Upgrade: Suspend Document Retrieval and Make Sure All Messages Have Been Sent

1. Go to **Packages > Management**. Click **View Locked Elements**, click **Unlock Elements**, select all elements of triggers, and then click **Unlock Selected Elements**.
2. Go to the **Settings > Messaging > Broker\Local Trigger Management** page. In the **Individual Trigger Controls** area, in the **Active** column under **Document Retrieval**, click **edit all** and set **Retrieval State** to **Suspended**. Refresh the page until the **Persisted Queue Counts** field shows 0 for every trigger.
3. Go to the **Settings > Resources > Store Settings** page. In the **Outbound Document Store** area, make sure the **Current Documents in Outbound Store** field shows 0.

9.5 Upgrade: Suspend Triggers and Make Sure Queues are Drained

1. Quiesce Integration Server. Specify at least 1 minute for the quiesce to occur, so Integration Server has time to stop executing new incoming requests and to finish executing in-flight services. For instructions and details about specific actions that occur when Integration Server is quiesced, see *webMethods Integration Server Administrator's Guide*.
2. Go to the **Settings > Messaging > webMethods Trigger Management** page. If the **Current Queue Counts** field does not show 0 for every trigger, diagnose and fix the problem (for example, the JMS provider might not be active or might be slow to process requests from Integration Server). Refresh the page until the **Current Queue Counts** field shows 0 for every trigger.
3. Go to the **Settings > Resources > Store Settings** page. In the **Outbound Document Store** area, make sure the **Current Documents in Outbound Store** field shows 0.

9.6, 9.7, 9.8, 9.9 Upgrade: Suspend Triggers and Make Sure Queues are Drained

1. Quiesce Integration Server. Specify at least 1 minute for the quiesce to occur, so Integration Server has time to stop executing new incoming requests and to finish executing in-flight services. For instructions and details about specific actions that occur when Integration Server is quiesced, see *webMethods Integration Server Administrator's Guide*.
2. Go to the **Settings > Messaging > webMethods Trigger Management** page. If the **Current Queue Counts** field does not show 0 for every trigger, diagnose and fix the problem (for example, the JMS provider might not be active or might be slow to process requests from Integration Server). Refresh the page until the **Current Queue Counts** field shows 0 for every trigger.
3. Go to the **Settings > Messaging > webMethods Messaging Settings** page. Make sure the **CSQ Count** field shows 0 for the Broker or Universal Messaging connection alias, depending on which product you are using.

Make Sure Queues are Drained and Complete or Suspend Business Processes

1. Go to the **Settings > Messaging > JMS Settings** page. In the **JMS Connection Alias Definitions** area, make sure the **CSQ Count** field shows 0 for every JMS connection alias.
2. If you are going to upgrade business processes, and you have business process models that use volatile transition documents and process instances of those models are in a Started state, allow the Started process instances to complete or suspend them before you upgrade.

Important: If you do not allow Started process instances to complete or suspend before you upgrade, you could lose data.

Note: Started process instances of models that use guaranteed transition documents do not have to complete or be suspended before you upgrade.

Prepare the Old My webMethods Server

When you edit configuration settings for My webMethods Server, you download the appropriate configuration file from the MywebMethodsServer database component to the My webMethods Server installation and make the changes in that file. You then either upload the file to the database component and delete it from the file system, or keep it in the file system so its settings are used in preference to the equivalent settings in the database. Go to the *old_Software AG_directory \MWS\server\instance_name \config* directory and check for any such files. If you do not want such files to be migrated to the new installation, delete them from the old installation.

Prepare the Old OneData

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

Go to *old_Software AG_directory \...*

Back Up This File

profiles \ODE \configuration \tomcat \conf

server.xml

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

Prepare the Old Optimize

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

Shut Down the Old Products

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

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

For the products below, follow these instructions.

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

Migrate Database Components

The new release might require changes to the database components, such as new tables, columns, keys, or indexes. In this section, you run 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. *new_Software AG_directory RDBMS product*

Shut Down Products that Connect to Database Components

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

Before You Migrate Trading Networks Database Components

1. Use your data editor to delete all data from the ARCHIVE_WORKTABLE table in the TradingNetworksArchive database component.

2. If you created custom indexes for your Trading Networks database components, check whether those custom indexes conflict with indexes that will be created when you run the Trading Networks database migration scripts in the next step. If so, drop the custom indexes. The database migration scripts are located in the *new_Software AG_directory \common\db\scripts\RDBMS \tradingnetworks\migrate* directory, under the directories listed below.

Old Release	Directories
9.0	\40_to_45 and 45_to_50
9.5 or 9.6	\45_to_50
9.7	\50_to_55
9.8	\55_to_56 and 56_to_65
9.9	\65_to_70

Special Considerations for Migrating Other Database Components

- If you are using business processes, you have stored information about in-progress, completed, or failed business process instances. This stored information is called business process runtime data. If you do not want to finish running in-progress process instances or resubmit completed or failed process instances from your old installation in your new installation, do not migrate your business process runtime data (see table below).
- If you are using Optimize with a DB2 RDBMS, there are special considerations for the URL you specify on the database component migration command. You must do the following:
 - Specify the schema name in the URL using all capital letters.
 - Specify the options `CreateDefaultPackage=true`, `ReplacePackage=true`, and `DynamicSections=3000`. These settings will affect all database components in the same schema or database.

Install Fixes on Database Migration Scripts

Install the latest fix on the database migration scripts. The fix names follow the convention `DC_old_release_number_DBS_Fixlatest_fix_number`. For instructions on installing fixes, see *Using the Software AG Update Manager* and the fix readme file.

Migrate Database Components

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

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 commands below. After you run each command, check the log file *dcc_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 an AIX system, enclose the values `"db_server_URL"`, `"existing_db_user"`, and `"password"` in quotation marks.

If you are upgrading...	Run this command...
Integration Server	<pre>dbConfigurator.{bat sh} -a migrate -d {oracle sqlserver db2luw} -pr IS -v latest -l db_server_URL -u existing_db_user -p password</pre> <p>Note: Migrate the Integration Server database components before migrating other database components.</p>
ActiveTransfer Server	<pre>dbConfigurator.{bat sh} -a migrate -d {oracle sqlserver db2luw} -c ActiveTransfer -v latest -l db_server_URL -u existing_db_user -p password</pre>
Business process runtime data and Business Rules	<pre>dbConfigurator.{bat sh} -a migrate -d {oracle sqlserver db2luw} -pr BPM -v latest -l db_server_URL -u existing_db_user -p password</pre>
Archive, Staging, or Reporting database components	<pre>dbConfigurator.{bat sh} -a migrate -d {oracle sqlserver db2luw} -c {Archive Staging Reporting} -v latest -l db_server_URL -u existing_db_user -p password</pre>
My webMethods Server	<pre>dbConfigurator.{bat sh} -a migrate -d {oracle sqlserver db2luw} -pr MWS -v latest -l db_server_URL -u existing_db_user -p password</pre>
OneData	<pre>dbConfigurator.{bat sh} -a migrate -d {oracle sqlserver} -c {OneDataMetadata OneDataReleaseArea OneDataWorkArea} -v latest -l db_server_URL -u existing_db_user -p password</pre>

If you are upgrading...	Run this command...
Optimize	<pre>dbConfigurator.{bat sh} -a migrate -d {oracle sqlserver db2luw} -pr OPTIMIZE -v latest -l db_server_URL -u existing_db_user -p password</pre>
Trading Networks	<pre>dbConfigurator.{bat sh} -a migrate -d {oracle sqlserver db2luw} -pr TN -v latest -l db_server_URL -u existing_db_user -p password</pre>

Prepare to Migrate Product Assets

Migration Utilities

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

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

- You can run a custom migration, in which you select the assets to migrate. The utility gathers your settings through a series of prompts, then migrates the selected assets. 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. 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_release sbs.dat`. The settings for default migrations are described in the product-specific chapters.

Note: When you run migration utilities, you provide the full path to the old installation, and sometimes the path to the 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.

If a migration utility fails to migrate an asset, the utility asks whether to continue with the next asset 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.

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 log4j.properties file (for example, the *new_Software AG_directory \product \bin \migrate* or *\migrate \bin* directory), open the file in a text editor, set the `log4j.logger.Migration` property to `DEBUG`, `MigrationFile`, and save and close the file.

Create Zip Files as Source of Assets to Migrate

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

Create a Zip File for My webMethods Server

1. On the old machine, open a command window or shell, go to the *old_Software AG_directory \MWS \bin \migrate* or *old_Software AG_directory \MWS \bin* directory, depending on which old release you have, and run the command `zip-mws.{bat|sh}`. The command creates a zip file named `mws.zip` in *old_Software AG_directory \MWS \migrate-zip* directory.
2. Copy the `mws.zip` file to any directory on the machine that hosts the new My webMethods Server.

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

Create Zip Files for All Other Products

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

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

1. Go to the product's old machine and open a command window or shell.
2. Go to the Software AG directory that contains the old product and enter the command indicated in the table below. If multiple commands are listed, use the same Zip file name in each command.
3. Copy the product's Zip file to any directory on the machine that hosts the new product.

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

Product	Run this command...
Application Platform	<p>You can reduce the size of the zip file by first moving the log files out of the <i>old_Software AG_directory</i> \profiles\product \logs directory, where <i>product</i> is the host Integration Server instance, the host My webMethods Server instance, or SPM (for Platform Manager).</p> <pre>jar cfM Zip_file_name.zip profiles</pre>
Command Central	<p>You can reduce the size of the zip file by first moving the log files out of the <i>old_Software AG_directory</i> \profiles\CCE\logs and \SPM\logs directories.</p> <pre>jar cfM Zip_file_name.zip common\conf jar uFM Zip_file_name.zip profiles\CCE jar uFM Zip_file_name.zip install\products jar uFM Zip_file_name.zip profiles\SPM</pre>
Event Routing	<pre>jar cfM Zip_file_name.zip install\products jar uFM Zip_file_name.zip common\conf jar uFM Zip_file_name.zip common\EventTypeStore jar uFM Zip_file_name.zip common\nerv\bundles</pre> <p>The common\nerv\bundles directory will exist only if you deployed NERV bundles in the old release.</p> <p>If you deployed NERV bundles or the Event Type Store to non-default locations, add the bundles to the Zip file to preserve the file structure shown in the commands above.</p>
Software AG Infrastructure (specifically, Software AG Runtime)	<p>You can reduce the size of the zip file by first moving the log files out of the <i>old_Software AG_directory</i> \profiles\CTP\logs directory.</p> <pre>jar cfM Zip_file_name.zip common\conf install\products profiles\CTP</pre> <p>You might see the message profiles\CTP: no such file or directory. You can ignore this message.</p>
Infrastructure Data Collector	<p>You can reduce the size of the zip file by first moving the log files out of the <i>old_Software AG_directory</i> \profiles\InfraDC\logs directory.</p> <pre>jar cfM Zip_file_name.zip common\conf install\products profiles\InfraDC</pre>
Integration Server	<p>9.0 and 9.5 upgrade: You can reduce the size of the zip file by first moving the log files out of the <i>old_Software AG_directory</i> \profiles\IS\logs and \IntegrationServer\logs directories.</p> <p>9.6, 9.7, 9.8, 9.9 upgrade: You can reduce the size of the zip file by first moving the log files out of the <i>old_Software AG_directory</i> \profiles\instance_name \logs, \IntegrationServer</p>

Product	Run this command...
	<pre>\instances\logs, and \IntegrationServer\instances \$instance_name \logs directories. jar cfM Zip_file_name.zip IntegrationServer*</pre>
OneData	<pre>jar cfM Zip_file_name.zip profiles\ODE\bin\onedata\config*</pre>
Universal Messaging	<p>9.0, 9.5, or 9.6 upgrade:</p> <pre>jar cfM Zip_file_name.zip install\products\NUMRealmServer.prop nirvana\server</pre> <p>9.7, 9.8, or 9.9 upgrade:</p> <pre>jar cfM Zip_file_name.zip install\products\NUMRealmServer.prop UniversalMessaging\server</pre>

Migrate the 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.

Before Migrating

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

Run the Software AG Infrastructure Migration Utility

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

The Software AG Infrastructure utility migrates the following:

- The Software AG Runtime debug, JAAS, JMX, port, security, single-sign on, watchdog, and Web Services Stack configurations; journal log settings; user repository; and password store. The utility also migrates any Java Service Wrapper customizations you made in the old wrapper_conf or custom_wrapper.conf files to the new custom_wrapper.conf file.

- 9.5, 9.6, 9.7, 9.8, or 9.9 upgrade: Users, groups, and roles.

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

For a general description of migration utility behavior, see "[Migration Utilities](#)" on page 39.

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

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Software AG_directory|Zip_file}
-destDir full_path_to_new_Software AG_directory
-importFile migrateold_release sbs.dat
-silent true
```

2. If you see a *Software AG_directory* \profiles\CTP directory in both the old and new installation directories, go to the *new_Software AG_directory* \profiles\CTP\bin \migrate directory and run the same command again.
3. Install the latest product fix on the new Common Platform and the new Security Infrastructure. Fix names for these products typically include the letters OSGI and SIN, respectively. For instructions on installing fixes, see *Using the Software AG Update Manager* and the fix readme files.

Migrate LDAP Server Configuration

Go to the *old_Software AG_directory* \profiles\CTP\configuration \com.softwareag.platform.config.propsloader\com.softwareag.security.ldap.server.pid-XXX.properties directory and copy all files to the *new_Software AG_directory* \profiles \CTP\configuration\com.softwareag.platform.config.propsloader directory.

Update the Software AG Runtime NERV Configuration

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

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

Migrate Java Service Wrapper Comments and #include Directives

If you added comments or #include directives to the old My webMethods Server Java Service Wrapper, copy them from the old wrapper_conf or custom_wrapper.conf

files to the new `custom_wrapper.conf` file. The files are located in the *Software AG_directory* \profiles\CTP\configuration directory.

Where to Go from Here

- If you see product-specific chapters for your products after this chapter, perform the tasks in those chapters in the exact order in which the chapters and tasks are presented in this guide. The order of performance is critical because your products have many inter-dependencies, including a shared infrastructure. Then go to ["Migrate Software AG Event Routing and Perform Final Upgrade Tasks for All Products" on page 119](#) and complete the tasks in that chapter.
- If you do not see product-specific chapters for your products after this chapter, go to ["Migrate Software AG Event Routing and Perform Final Upgrade Tasks for All Products" on page 119](#) and complete the tasks in that chapter.

4 Migrate Terracotta, Universal Messaging, and webMethods Broker

■ Read and Complete All Previous Chapters	46
■ Migrate Terracotta	46
■ Migrate Universal Messaging	46
■ Migrate webMethods Broker	49

Read and Complete All Previous Chapters

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

Migrate Terracotta

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

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

You must upgrade all Terracotta Server Array clients and servers in the cluster to 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 Universal Messaging

Before Migrating

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

Run the Universal Messaging Migration Utility

The Universal Messaging migration utility migrates server instances within the old installation to the new installation. This includes migrating each old instance's data directory to the corresponding new instance. The migration utility also creates instances

of Template Applications and Enterprise Manager, if you installed them in the new installation.

For a general description of migration utility behavior, see ["Migration Utilities" on page 39](#). The Universal Messaging migration utility does not offer a default migration.

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.
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 Custom Imported Settings with Prompting

Imported settings come from 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.

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.

9.0 through 9.7 Upgrade: 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 migrate.dat
-silent true
```

9.8 and 9.9 Upgrade: Migrate without Using Custom Imported Settings or 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}
-instanceName instance_name [, instance_name , instance_name ...]
-importFile NONE
-silent true
```

Migrate Custom Changes and Data Directory

- 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.
 - 9.0 upgrade: If you made any custom changes to the `nserver.conf` or `nserverdaemon.conf` file for an old instance, make the same changes to the `Server_Common.conf` file for the corresponding new instance. The `nserver.conf` and `nserverdaemon.conf` files are stored in the *old_Software AG_directory \nirvana \server \nirvana \bin* directory.
 - 9.5, 9.6, or 9.7 upgrade: If you made any custom changes to the `nserver.conf` or `nserverdaemon.conf` file for an old instance, make the same changes to the `Server_Common.conf` file for the corresponding new instance. The `nserver.conf` and `nserverdaemon.conf` files are stored in the *old_Software AG_directory \nirvana \server \umserver \bin* directory.
 - 9.8 or 9.9 upgrade: If you made any custom changes to the `Server_Common.conf` file for an old instance, make the same changes in the `Server_Common.conf` file for the corresponding new instance.
- 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. In you copy it, edit the `DDATADIR` parameter in the `Server_Common.conf` file for the new instance to point to the new data directory location.

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

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

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

Migrate webMethods Broker

Types of Migration

Migrate from Old webMethods Broker to New Universal Messaging

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

Migrate from Old webMethods Broker to New webMethods Broker

In the 9.7 and later webMethods releases, the webMethods Broker release number is still 9.6 and is the same release that was available with the webMethods 9.6 release. For most types of upgrade, the tasks you perform depend on whether the new machine is the same as or compatible with the old machine, where “compatible” means the new machine has a similar hardware architecture that is running on the same or a higher version of the same operating system.

Type of Upgrade	Instructions
9.6, 9.7, 9.8, or 9.9 upgrade: The old webMethods Broker resides on its own machine, and you are going to continue using it with the new webMethods release.	Point 9.10 Broker clients, including Integration Server 9.10, to the old webMethods Broker.
9.0 or 9.5 upgrade: You installed the new webMethods Broker on an IBM System z machine that is running SUSE Linux Enterprise Server or Red Hat Enterprise Linux.	Create new storage and then migrate the Broker Server configuration from the old storage to the new storage using the webMethods Broker export \import utilities. For instructions, see <i>Administering webMethods Broker 9.6</i> .

Type of Upgrade	Instructions
You installed the new webMethods Broker on a machine that is not compatible with the old machine.	Create new storage and then migrate the Broker Server configuration from the old storage to the new storage using the webMethods Broker export \import utilities. For instructions, see <i>Administering webMethods Broker 9.6</i> .
You installed the new webMethods Broker on the same machine as the old webMethods Broker, or on a machine that is compatible with the old machine.	Follow the instructions in Create a Broker Server and Use the Old Storage , below.

Create a Broker Server and Use the Old Storage

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

```
server_config remove full_path_to_old_data_directory
```

When asked whether to remove the Broker Server from the Broker Monitor configuration, enter Y.

5. If you need to copy the data directory to a new location (for example, because the old and new webMethods Broker installations are on different machines), do the following:
 - a. Copy the old data directory to the new location.

You might have old storage files (for example, .stor and .data files) that were located outside the old data directory. You can copy these files to the new data directory, or you can copy these files to a location outside the new data directory.
 - b. On the machine that hosts the new webMethods Broker, open a command window or shell, go to the *new_Software AG_directory* \Broker\bin directory, and run the command below. The command configures the Broker Server to use the new data directory paths for the old storage files.

If you copied old storage files to a location outside the new data directory, use the `-qs_map_file` option for each old storage file to map the file's location on the old machine to the file's location on the new machine.

```
server_config relocate full_path_to_new_data_directory
[-qs_map_file full_path_to_old_storage_file_in_old_location
full_path_to_copy_of_storage_file_in_new_location]
```

6. If you are migrating from and to multi-home machines (that is, machines that have multiple network interfaces and IP addresses), open the awbroker.cfg file and update the broker-ipaddress configuration to point to the new machine. If you do not, you might experience a port binding error when you create the Broker Server.
7. When you create a Broker Server, you will specify the same host name and ports for the new Broker Server that you used for the old Broker Server. If you installed the new webMethods Broker on the same machine as the old webMethods Broker, the ports will be available because you shut down the old webMethods Broker earlier in this procedure. If you installed the new webMethods Broker on a different machine than the old webMethods Broker, however, the ports might be in use by another application. Make sure the ports are available.
8. Create a Broker Server. On the machine that hosts the new webMethods Broker, open a command window or shell, go to the *new_Software AG_directory* \Broker \bin directory, and run the appropriate command below. Broker Server details are automatically added to the Broker Monitor startup configuration. Specify the same host name and ports for the new Broker Server that you used for the old Broker Server.

Location of Old Data Directory	Command
Old location	<pre>server_config add full_path_to_old_data_directory -k 9.6_license_file -p old_release_port</pre>
New location	<pre>server_config add full_path_to_new_data_directory -k 9.6_license_file -p old_release_port</pre>

9. Make sure the new Broker Server is running, as follows:

System	Steps
Windows	<ol style="list-style-type: none"> a. Open the Windows Services window and make sure the status of the Software AG Broker Server 9.6(port) service is Started. b. Right-click the service and click Properties. Make sure the Path to Executable field points to the awbroker file for 9.6.
UNIX	<ol style="list-style-type: none"> a. Run the command <code>ps -ef grep awbroker</code>, which prints all running Broker processes including the path to the executable. Make sure the output text shows a line like this: <pre>full_path_to_new_Software AG_directory\Broker\bin \awbroker -d full_path_to_new_data_directory</pre> b. Make sure the awbroker process is running and that it is running from the 9.6 awbroker file.

5 Migrate My webMethods Server

■ Read and Complete All Previous Chapters	54
■ Before Migrating	54
■ Run the My webMethods Server Migration Utility	54
■ Migrate Java Server Wrapper Comments and #include Directives	56
■ Initialize a My webMethods Server Instance	56
■ Switch from Clone to Live Database	57

Read and Complete All Previous Chapters

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

Before Migrating

1. If you are upgrading a My webMethods Server cluster, read "[My webMethods Server Requirements](#)" on page 22.
2. If the new and old My webMethods Servers are on the same machine, make sure the old My webMethods Server is shut down.

Run the My webMethods Server Migration Utility

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

For a general description of migration utility behavior, see "[Migration Utilities](#)" on page 39. The `log4j.properties` file for My webMethods Server is located in the `new_Software AG_directory \MWS\bin\migrate\resources` directory.

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 whether to migrate the instance, and the database to use with the instance. If you respond that you are using a clone of the database, the utility prompts for the database URL, user name, and password.

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

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

6. The utility asks whether to begin migration. If you enter Y, the utility migrates the instances you specified. If migration completes successfully, the utility also does the following:
 - Deletes old configuration files that are not used by the new My webMethods Server, and deletes old properties that are not used by the new My webMethods Server from the new configuration files.
 - If the old My webMethods Server used the embedded database, copies the database tables from the old My webMethods Server to the new My webMethods Server and converts the tables to the new format.
 - Synchronizes the My webMethods Server installation directory with the database.

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_release sbs.dat` 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 and the Live Database 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 you want to migrate all instances, do not specify `-instanceName`. For the `migrateold_release` sbs.dat file, specify the source release number without periods (for example, 980).

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}
[-instanceName instance_name [, instance_name , instance_name ...]]
-importFile migrateold_release sbs.dat
-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
```

Migrate Java Server Wrapper Comments and #include Directives

If you added comments or `#include` directives to the old My webMethods Server Java Service Wrapper, copy them from the old `wrapper_conf` or `custom_wrapper.conf` files to the new `custom_wrapper.conf` file. The files are located in the `Software AG_directory \profiles\MWS_instance_name \configuration` directory.

Initialize a My webMethods Server Instance

The initializing process might take several hours.

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

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

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

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

- a. Log on to one of the new My webMethods Server instances as Administrator and go to the **Administration > My webMethods > Cluster Settings > Advanced Web and Cluster Configuration for MWS** page.
- b. If the host name is not correct in the **Host** and **MWS Front End URL** field, update the host name.
- c. Go to the **Cluster Status and Control** page and restart the instance.

Switch from Clone to Live Database

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

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

6 Migrate Integration Server and Hosted Wm Packages

■ Read and Complete All Previous Chapters	60
■ Packages	60
■ Before Migrating	60
■ Run the Integration Server Migration Utility	61
■ Complete the Integration Server Upgrade	65
■ Complete the ActiveTransfer Upgrade	67
■ Complete the CloudStreams Upgrade	68
■ Complete the Trading Networks Upgrade	69
■ Complete Adapter Upgrades	69
■ Complete eStandards Module Upgrades	71

Read and Complete All Previous Chapters

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

Packages

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

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

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

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

Before Migrating

1. If you are upgrading an Integration Server cluster, read "[Integration Server Clustering Requirements](#)" on page 21.
2. If you are upgrading an Integration Server that hosts Deployer, read "[Migrate the Asset Build Environment, Deployer, and Application Platform](#)" on page 89 and complete all necessary pre-migration tasks.
3. If the new and old Integration Servers are on the same machine, make sure the old Integration Server is shut down.
4. The migration utility will scan the old installation for custom packages. However, it will not find custom packages whose names start with Wm, as this naming convention is used for packages provided by Software AG. If you have custom packages whose names start with Wm, and you want to migrate them, go to the

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.

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

Run the Integration Server Migration Utility

The Integration Server migration utility can migrate the following:

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

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

For a general description of migration utility behavior, see "[Migration Utilities](#)" on page 39.

Perform a Custom Migration

1. On the machine that hosts the new Integration Server, 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 Integration Server installation or the Zip file you made earlier.
4. 9.6, 9.7, 9.8, or 9.9 upgrade: If the old Integration Server installation has more than one server instance, the utility asks which instance to migrate and lists the instances in the old installation.

The utility then asks for the name of a new instance to be the target of the migration. If you want the utility to migrate to the new instance that was created during installation, enter the name of that instance. If you want the utility to create a new instance and migrate to the 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.

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

5. The utility asks whether to import migration settings. Enter N.
6. The utility asks whether to migrate custom packages. If you choose to migrate all custom packages, the utility migrates the following:
 - Custom packages.
 - The WmChemPayloads, WmPapinetPayloads, WmRNPIps, WmFINMessages, and WmFIXMessages packages, if present. These packages contains IS documents for the corresponding eStandards Modules and the schemas for those documents.

If you choose to migrate selected packages only, the utility lists each of the packages above and asks whether to migrate it.

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

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

If the Integration Server hosts Trading Networks, and you are using a cloned database with the new Trading Networks, you must respond N to all prompts that list the Trading Networks JDBC files TN.xml and *pool_alias_name_in_TN.xml*. If you are using the live database with the new Trading Networks, respond Y to these prompts.

9. The utility asks you to specify the behavior to use for new properties that have been added to the new Integration Server and existing properties that have new defaults in the new Integration Server. If you choose to specify the behavior to use, the utility lists each property and asks whether to use new behavior or preserve existing behavior. You can also read about these properties in the Integration Server readme.
10. If the Integration Server hosts Wm packages, the utility asks whether to migrate the data (that is, configuration files) for those packages. Examples of hosted Wm packages are ActiveTransfer (WmMFT), Mediator (WmMediator), Process Engine (WmPRT), adapters, and eStandards Modules.

11. 9.5, 9.6, 9.7, 9.8, 9.9 upgrade: The utility asks whether to migrate custom jar files. If you choose to migrate selected custom jar files only, the utility lists each custom jar file and asks whether to migrate it.
12. 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.
13. 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.
14. The utility asks whether to begin migration. If you enter Y, the utility migrates the data you selected. If migration completes successfully, the utility also performs the actions in ["Other Actions Performed by the Migration Utility" on page 64](#).

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. For each old release, the settings are stored in a file named migrateold_release.sbs.dat file provided by Software AG in the *new_Software AG_directory* \IntegrationServer\bin\migrate directory. The settings tell the migration utility to migrate the data listed under ["Run the Integration Server Migration Utility" on page 61](#). For Integration Server

properties that are new or that have new defaults, the default migration chooses the behavior that best preserves backwards compatibility.

Migrate Using Custom or Default Imported Settings with Prompting

1. On the machine that hosts the new Integration Server, 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 Integration Server installation or to the Zip file you made earlier.
3. 9.6, 9.7, 9.8, or 9.9 upgrade: If the old Integration Server installation has more than one server instance, the utility lists the instances and asks which instance to migrate from.
4. The utility asks for the name of a new server instance to be the target of the migration. See this step in ["Perform a Custom Migration" on page 61](#) for detailed information on what to enter and how the utility proceeds.
5. The utility asks whether to import migration settings. Enter Y and, when prompted, provide the full path to the `migrate.dat` file or the `migrateold_release sbs.dat` file. For the `migrateold_release sbs.dat` file, specify the source release number without periods (for example, 980).

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

On the machine that hosts the new Integration Server, 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_release sbs.dat` file, specify the source release number without periods (for example, 980).

The `-instanceName` option specifies both the old server instance to migrate and the new server instance to migrate to; if the new instance does not exist, the utility will create it. If an error occurs, the utility exits.

```
migrate.{bat|sh}
{-srcDir|-srcFile} full_path_to_{old_Integration
Server_directory|Zip_file}
-instanceName instance_name
-importFile {full_path_to_migrate.dat | migrateold_release sbs.dat}
-silent true
```

Other Actions Performed by the Migration Utility

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

- Deletes old configuration files that are not used by the new Integration Server, and deletes old properties that are not used by the new Integration Server from the new configuration files.

- If the old Integration Server used the embedded database, the utility copies the database tables from the old Integration Server to the new Integration Server and upgrades the tables to the new format if necessary.
- Adds a new property named "Validate schemas using Xerces" to existing Web service descriptors, and sets the new property to the value the `watt.server.wsdl.validateWSDLSchemaUsingXerces` parameter had in the old Integration Server. The new property replaces the functionality provided by that parameter. For information about the new property, see *webMethods Service Development Help*.
- If you are migrating Business Rules data, the utility upgrades Business Rule projects. You might see XML parsing messages due to a Java bug; you can ignore these messages.
- If you are upgrading CloudStreams, the utility migrates configuration artifacts related to administering CloudStreams Server.

Complete the Integration Server Upgrade

Migrate Java Server Wrapper Customizations

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

Customize Integration Server Startup Files

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

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.
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 and point to the new Integration Server.
- b. Go to the **Settings > Extended** page. If you have the extended setting `watt.server.xml.ncname.encode.backward.compatibility` and it is set to true, reset it to false.
- c. Regenerate the clients for all Provider Web services that have an operation with field names starting with `xml` in the input signature, output signature, header, or faults.

Synchronize and Update Serial webMethods Messaging Triggers

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

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

Connect Integration Server Functions to Database Components

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

Update Host Names

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

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

Database Component	Table	Column
ISInternal	IS_USER_TASKS	TARGET

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

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

Update the Integration Server NERV Configuration

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

9.0 Upgrade: Migrate Custom Jar Files

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

Complete the ActiveTransfer Upgrade

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

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

Complete the CloudStreams Upgrade

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

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

The input signature for the service is as follows:

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

Parameter	Description
	<ul style="list-style-type: none"> ■ <code>false</code> to migrate CloudStreams artifacts from only custom packages specified on the <code>packages</code> parameter. This is the default.
<code>packages</code>	Object. String array containing the names of custom packages to migrate. You must specify at least one package name.

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

Field	Description
<code>packageName</code>	String. Name of the custom package.
<code>success</code>	String. Value that indicates whether migration succeeded (<code>true</code>) or failed (<code>false</code>).
<code>message</code>	String. 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>	Object. Optional. If the service issued info messages during the migration, string array of the messages.
<code>errors</code>	Object. Optional. If the service issued errors during the migration, string array of the errors.

Complete the Trading Networks Upgrade

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

Complete Adapter Upgrades

If the old Integration Server hosted adapters and you installed the same release or a later release of those adapters on the new Integration Server, you migrated data for those

adapters as part of the Integration Server migration. For the adapters below, however, you must also complete the indicated tasks. Then restart Integration Server.

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

webMethods Adapter for JDBC

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

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

webMethods Package for Microsoft .NET

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

webMethods PeopleSoft EnterpriseOne Adapter

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

webMethods SAP Adapter

- 9.0 upgrade: Go to the *old_Software AG_directory* \IntegrationServer\lib\jars directory and copy all files whose file name contains the phrase `*sapjco*.*` to the *new_Software AG_directory* \IntegrationServer\instances*instance_name* \lib\jars \custom directory.
- Go to the specified directory below:

Old Installation	Directory
9.0 or 9.5	<i>old_Software AG_directory</i> \IntegrationServer\packages \WmSAP\ns\wm\sap\internal\la\routingListener
9.6, 9.7, 9.8, or 9.9	<i>old_Software AG_directory</i> \IntegrationServer\instances \instance_name \packages\WmSAP\ns\wm\sap\internal \la\routingListener

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

- If the new Integration Server is on a UNIX system, set these environment variables on the machine:

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

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

Complete eStandards Module Upgrades

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

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

webMethods SWIFTNet Module

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

7 Migrate Software AG Designer and Business Process Runtime

■ Read and Complete All Previous Chapters	74
■ Migrate Software AG Designer	74
■ Migrate the Business Process Runtime	77
■ Migrate Closed Loop Analytics	78

Read and Complete All Previous Chapters

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

Migrate Software AG Designer

Before Migrating

1. Open the new Software AG Designer and point to a new workspace. For example, you can accept the default workspace *release*.
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 4.5, so make sure any features you add are compatible with that Eclipse release.
3. If you are upgrading Events, you exported your Event Bus Console configurations in "[Export Event Bus Console Configurations](#)" on page 32. Import the configurations as follows:
 - a. Go to the **File > Import** wizard.
 - b. In the Select panel, go to **Run\Debug > Launch Configurations** and click **Next**.
 - c. In the **From Directory** field, go to the directory to which you exported the configurations and select all files with the extension **.launch**. Click **Finish**.
4. If you exported your preferences before upgrading, import them as follows:
 - a. Go to the **File > Import** wizard. In the Select panel, go to **General > Preferences** and click **Next**. In the **From preference file** field, specify the **.epf** file to which you exported your preferences. Click **Finish**.
 - b. Go to **Window > Preferences**. In the preferences window, go to the **Java > Installed JREs > Execution Environments** page, if multiple JREs are listed, make sure JRE 1.8 is selected as the default, or remove older JREs from the list. Also update settings that point to old product installations to point to new product installations instead. For example, for My webMethods Server, update the **Server > Runtime Environments** settings. For Application Platform, update the path to the runtime instance, then review the messages on the Error tab to see whether any other paths need to be updated.
 - c. Restart Software AG Designer.

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

1. For 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. For Apama, use **Project > Clean** to force a rebuild of generated project artifacts under Software AG Designer's control.
4. Verify the following:

Product	Verify that...
Apama	All projects build without errors. If errors appear in the Problems tab, click each error and resolve it.
Application Platform	All projects compile without errors. If errors appear in the Problems tab, click each error and resolve it. For example, if you see errors indicating that classes cannot be resolved to a type, edit the project's class path configuration.
Business Process	All imported business process projects appear in the Solutions tab under the Processes node.
Business Rules	All imported business rules projects appear in the Solutions tab under the Rules node.

5. For Apama, if you use ant to automate building and deploying your projects, export your ant deployment scripts to an empty directory. Run the scripts to force a rebuild of any generated artifacts within your projects (for example, Apama queries,

scenarios or Java\C\C++\.NET code using Apama APIs). Review the build output for errors and warnings.

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

Migrate Task Application Projects

1. In the new Software AG Designer, 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.
2. Go to **Software AG > Task Development**. In the preferences window, make sure all your other task-related Software AG Designer preferences are correct, and then click **OK**.
3. Go to the **File > Import** wizard.
 - a. In the Select panel, go to **Software AG > Existing CAF Projects into Workspace** and then click **Next**.
 - b. In the Import Projects panel, click **Select Root Directory** and go to the workspace or source control system that contains your task application projects, or click **Select Archive File** and go to the directory that contains your task application projects. In the **Projects** box, select the projects to import.
 - c. Select **Copy projects into workspace** and then click **Finish**.
4. Make sure all imported task application projects appear in the **Solutions** tab, under the Tasks node.
5. If errors appear in the **Problems** tab, click the **Navigator** tab. Right-click each task application project, click **CAF Tools**, and click **Repair CAF Project**. If errors still appear, restart Software AG Designer.
6. Publish the migrated task application projects to 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. Start the new Software AG Designer.

2. Import CloudStreams Server definitions as follows:
 - a. Go to the **CloudStreams Development** perspective, then go to **Window > Preferences**. On the **Preferences** dialog box, in the left navigation bar, go to **Software AG > CloudStreams Servers**.
 - b. Click **Import**, select the .properties file you exported, and click **Open**. Software AG Designer asks whether to overwrite existing servers. Click **OK**, and then click **OK** again to close the **Preferences** dialog box.
3. Import CloudStreams Governance projects as follows:
 - a. Go to the **File > Import** wizard. In the Select panel, go to **Software AG > CloudStreams Governance Project** and then click **Next**.
 - b. In the Import CloudStreams Governance Project panel, specify the directory that contains the projects you exported in the **Select Root Directory** field. In the **Projects** box, select the projects to import.
 - c. Select **Copy projects into workspace** and then click **Finish**.
 - d. In the **CloudStreams Development** perspective, make sure all imported governance projects appear in the **CloudStreams Governance** tab.
4. Repeat the previous step to import CloudStreams Provider projects.

Migrate Integration Server Definitions

1. In the new Software AG Designer, go to **Window > Preferences**. On the **Preferences** dialog box, in the left navigation bar, go to **Software AG > Integration Servers**.
2. Click **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 the Business Process Runtime

Note: 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.

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

that connects to webMethods Broker. Go to the **IS Settings > Messaging > webMethods Messaging Settings** page and check the Broker name.

2. If you migrated process runtime packages (see "[Run the Integration Server Migration Utility](#)" on page 61), 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. If your old and new product installations are on different machines, the new Integration Server that hosts Monitor is not connected to any messaging product and has no process model fragments, and you want to be able to resubmit processes and process steps that ran before you upgraded, follow the steps below.
 - a. In Integration Server Administrator, go to the **Package > Management** page and click  for the WmMonitor package.
 - b. Clear the **Resubmit to local IS** check box, click **Submit**, and then reload the package.
 - c. In your database editor, go to the PRA_PROCESS_STEP database table in the ProcessAudit database component and update the SERVERID column to point to the new Integration Server that hosts Monitor.
5. Make sure document retrieval for all webMethods Messaging triggers is enabled on new Integration Servers. For instructions, see *webMethods Integration Server Administrator's Guide*.

Migrate Closed Loop Analytics

In "[Migrate Software AG Designer](#)" on page 74 you migrated all assets involved in Closed Loop Analytics. You must now update the JMS topic in Universal Messaging from which Closed Loop Analytics collects escalation events, as follows:

1. Start the Universal Messaging instance from the new release that you are using for BPM, and then start the Enterprise Manager.
 - In the left panel, open the Realms tree and click the Universal Messaging instance you are using for BPM.
 - In the right panel, click the JNDI tab and open the Topics tree. Delete the topic for the escalation event type {http://namespaces.softwareag.com/EDA/WebM/Rules/CLA/1.0}ProcessEscalation.
2. In the file system of the machine that hosts the new Universal Messaging instance, open a command line window and create a new topic for the escalation event type by entering the command below. Type the command on one line with a space at each of the line breaks shown below.

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

8 Migrate Optimize and Infrastructure Data Collector

■ Read and Complete All Previous Chapters	82
■ Configure and Start the Terracotta Server Array	82
■ Update Connection to Other Products	82
■ Migrate Infrastructure Data Collector	83
■ Deploy the Optimize Environment	85
■ Switch from webMethods Broker to Universal Messaging	87

Read and Complete All Previous Chapters

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

Configure and Start the Terracotta Server Array

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

Update Connection to Other Products

Update Connection to Process Engine

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

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

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

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

Update Connection to Optimize Support Package

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

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

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

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

Migrate Infrastructure Data Collector

Run the Infrastructure Data Collector Migration Utility

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

For a general description of migration utility behavior, see "[Migration Utilities](#)" on page 39.

Before Migrating

1. If the new and old Infrastructure Data Collectors are on the same machine, make sure the old Infrastructure Data Collector is shut down.
2. If you want to migrate SNMP asset configuration files, go to the *new_Software_AG_directory* `infrastructuredc\bin\migrate\old_release` directory and open the `snmpMigration.properties` file. Enter the full paths to the old and new SNMP configuration directories. The contents of the file show the format to use.

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_release sbs.dat` file in the `new_Software AG_directory \infrastructuredc \bin \migrate` directory. The settings tell the migration utility to migrate the data listed in "[Migrate Infrastructure Data Collector](#)" on page 83.

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_release sbs.dat` file. For the `migrateold_release sbs.dat` file, specify the source release number without periods (for example, 980).

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_release* sbs.dat file, specify the source release number without periods (for example, 980).

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 | migrateold_release sbs.dat}
-silent true
```

Migrate Java Server Wrapper Comments and #include Directives

If you added comments or #include directives to the old My webMethods Server Java Service Wrapper, copy them from the old wrapper_conf or custom_wrapper.conf files to the new custom_wrapper.conf file. The files are located in the *Software AG_directory* \profiles\InfraDC\configuration directory.

9.7 Upgrade: Start Monitoring 4.1 or Later Terracotta Server Arrays

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

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

Deploy the Optimize Environment

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

Not Reuse the Old Optimize CCS Environment Definition

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

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

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

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

Reuse the Old Optimize CCS Environment Definition

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

1. Start the new My webMethods Server and open My webMethods.
2. Import the old CCS environment, as follows:
 - a. Go to the **Applications > Administration > System-Wide > Environments > Define Environments** page.
 - b. Click **Import Environment**. On the Import Environments dialog box, click **Browse**, go to the file to which you exported the old CCS environment, and click **OK**. When prompted, confirm the migration.
 - c. When asked "Do you wish to migrate the environment to the latest version?", click **Yes**. The CCS imports the old environment and migrates it to the new Optimize Logical Servers.
3. Update the database pool connection settings for the database pool definition associated with the environment you imported, as follows:
 - a. Go to the **Applications > Administration > System-Wide > Environments > Database Pool Configuration** page.
 - b. In the **Pool** column, click the imported database pool.
 - c. In the **Pool Settings** area, set the **Minimum Connections** field to 8, and the **Maximum Connections** field to 60.
 - d. Make sure the database connection information is correct. In the **Database Connection** area, click **Test** and make sure the message `Test Passed` displays at the bottom of the page.
 - e. Click **Save**.
4. Go to the **Applications > Administration > System-Wide > Environments > Define Environments** page. Under **Environment Name**, click the environment you imported.
5. If you are using a different JMS provider with the new release than you used with the old release, do the following:

- a. Click the **Configure Servers** tab. Under **Configuration**, open the **Default Settings** tree and click **JNDI Configuration**.
 - b. In the **Naming Factory Type** list, click the JMS provider you are using with the new release. If you are using webMethods Broker, complete the **Broker Name** field. If you are using Universal Messaging, clear the field.
 - c. Click the **Map Endpoints** tab. In the first **Incoming Connections** field for the JMS Provider, type `broker` if you are using webMethods Broker or `nsp` if you are using Universal Messaging. In the second field, type the port number.
 - d. Click **Save**.
6. Validate the environment as follows:
 - a. Click each tab that shows the icon  except the **Validate** tab. If you change any settings on a tab, and the tab has a **Save** button, click **Save**.
 - b. When all tabs except the **Validate** tab show , click the **Validate** tab.
 7. Click **Finish**.

Deploy the New Optimize Environment

1. Start the new Optimize components you installed. These components can include the Analytic Engine, Infrastructure Data Collector, and the Web Services Data Collector.
2. Start the JMS provider 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. You can identify the hosts using DNS name or IP address.
 - If you have BPMS installed, but are not using BPM (that is, you are using only Optimize), select the **BAM only** option and identify the new Analytic Engine. You can identify the host using DNS name or IP address.

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

Switch from webMethods Broker to Universal Messaging

If you were using webMethods Broker as your JMS provider in the old release, and are using Universal Messaging in 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.

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

■ Read and Complete All Previous Chapters	90
■ Migrate the Asset Build Environment	90
■ Migrate Deployer	90
■ Migrate Application Platform	93

Read and Complete All Previous Chapters

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

Migrate the Asset Build Environment

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

Migrate Deployer

You have two options for migrating Deployer data:

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

The instructions for each option are provided below.

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

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

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

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

3. Make sure you are a member of the Administrators ACL, or of the Internal, Developer ACLs, and DeployerAdmin ACLs, for both the Integration Server that hosts the old Deployer and the new Integration Server that hosts the new Deployer. For instructions, see *webMethods Integration Server Administrator's Guide*.
4. 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.
5. If you installed the new Deployer on the same machine as the old Deployer, you might want the new Deployer to use the ports that were used by the old Deployer. If so, do the following:
 - a. In Integration Server Administrator, point to the Integration Server that hosts the old Deployer and go to the **Security > Ports** page.
 - b. Click **Add Port**, click **webMethods\HTTP**, and click **Submit**. Enter the port details, click **Yes** for **Enable**, click **WmRoot** under the package name, and click **Save Changes**.
 - c. Click **Change Primary Port**, click the new port, and click **Update**.
 - d. In the **Access Mode** column of the new port, click **Edit**. In the **Edit Access Mode** page for the port, click **Set Access Mode to Allow by Default**. Click **OK** on the warning pop-up. The message `Successfully changed access mode for Port Service Access Settings` appears.
 - e. Disable the old port by clicking **Yes** in the **Enabled** column for that port.

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

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

Migrate Deployer settings, server aliases, target groups, and projects using the instructions in ["Migrate Integration Server and Hosted Wm Packages" on page 59](#).

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

Start the Old and New Deployers and Migrate Global (Default) Settings

1. Start the old Deployer and the new Deployer.

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

Migrate Server Aliases (Connection Information)

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

1. In the new Deployer, click **Migrate Server Aliases**. In the **Select Server** list, click the type of server whose aliases to migrate.
2. In the **Select** column, select the check boxes next to the server aliases you want to migrate. In the **Point Selected Aliases to Servers** list, click **9.10**.
3. Click **Migrate Server Aliases**. Deployer migrates the selected server aliases, then disables those rows and displays **Yes** in the **Migrate** column for the server aliases.
4. Repeat the steps above for each type of server whose aliases to migrate.
5. You cannot migrate Integration Server HTTPS server aliases from Deployer. Manually recreate these server aliases in the host Integration Server.

Migrate Target Groups

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

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

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

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

4. In the **Select** column, select the check boxes next to target groups to migrate. In the **Point Selected Aliases to Groups** list, click **9.10**.
5. Click **Migrate Target Groups**. Deployer migrates the selected target groups, then disables those rows and displays **Yes** in the **Migrate** column for the groups.

Migrate Deployment Projects

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

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

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

Migrate Application Platform

1. Rebuild and redeploy your Application Platform projects using Asset Build Environment and Deployer. For instructions, see *webMethods Application Platform User's Guide*.
2. Run the Application Platform migration 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_release sbs.dat* file, specify the source release number without periods (for example, 980).

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


10 Migrate Apama

■ Read and Complete All Previous Chapters	96
■ Before Migrating	96
■ Complete the Apama Upgrade	96

Read and Complete All Previous Chapters

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

Before Migrating

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.

If the new and old Apama are on the same machine, make sure all processes from the old installation are shut down.

Complete the Apama Upgrade

1. Run your ant scripts to force a rebuild of any generated artifacts within your projects (for example, Apama queries, scenarios or Java\C\C++\.NET code using Apama APIs). Check the build output for errors and warnings.
2. 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.

11 Migrate API-Portal and CentraSite

■ Read and Complete All Previous Chapters	98
■ Migrate API-Portal	98
■ Migrate CentraSite	99

Read and Complete All Previous Chapters

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

Migrate API-Portal

1. If the new and old API-Portals are on the same machine, make sure the old API-Portal is shut down.
2. If you installed the new API-Portal in a clustered, high-availability setup, make sure all nodes are running and accessible to the Zookeeper ensemble. Then register all the nodes with the parent node by invoking the REST service shown below. The default for the load balancer port is 18004.

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

For	Run these commands...
9.7	<pre>acc> set acc config backup.restore.tenant.app.types = UMC acc> restore tenant master from full_path_to_backup_file username=your_user_name password=your_password</pre>
9.8	<pre>acc> set acc config backup.restore.tenant.app.types=ECP,ADS,UMC acc> restore tenant master from full_path_to_backup_file username=your_user_name password=your_password</pre>
9.9	<pre>acc> set acc config backup.restore.tenant.app.types=ECP,ADS,UMC acc> restore tenant master from full_path_to_backup_file username=your_user_name password=your_password</pre>

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

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

- In the User Management Component at `http://host:port/umc`, import the license for each tenant you created.
- In the new API-Portal Cloud Controller, run these commands:

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

- 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.

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

- Run these commands:

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

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

Migrate CentraSite

Before Migrating

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

System	Command
Windows	<code>sbsExport.cmd full_path_to_zip_file</code>
UNIX	<code>sbsExport.sh \ full_path_to_zip_file</code>

An example of this command for UNIX is as follows:

```
.\sbsExport.sh \tmp\sbs_cs82_data.zip
```

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

Migrate the Data

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

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

```
sbsImport.{cmd|sh} \full_path_to_zip_file
```

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

2. Start the new CentraSite.
3. 9.0, 9.5, or 9.6 upgrade: Transform and migrate the LDAP configuration to the new JAAS configuration. From the *new_Software AG_directory \CentraSite\utilities* directory, run this command:

```
CentraSiteCommand.{cmd|sh} generate JaasConfiguration -url CentraSite_new_URL  
-user CentraSite_new_admin_user -password password
```

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

4. If you use single sign-on with CentraSite, do the following:
 - a. Open the `jaas.config` file in the old and new *Software AG_directory \CentraSite \profiles\CTP\configuration* directories. Copy the following from the old file to the 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.

Complete the Upgrade

1. Change any settings in CentraSite Control that you noted in "[Prepare the Old CentraSite](#)" on page 31.
2. If you installed plug-ins that are GUI extensions for CentraSite Control in the old CentraSite installation, install them in the new CentraSite installation.
3. If you are using API-Portal with CentraSite, republish all API-Portal instances you created in the old release to the new API-Portal. For instructions, see the CentraSite documentation.

12 Migrate MashZone NextGen

■ Read and Complete All Previous Chapters	104
■ Overview	104
■ Set Up the MashZone NextGen Repository	104
■ Update the MashZone NextGen Configuration and Copy Extensions	105
■ Copy Integrated MashZone Data and Run Upgrade Commands	106
■ 9.9 Upgrade: Upgrade JDBC Configuration and Run Upgrade Commands	108
■ 3.6 or 3.7 Upgrade: Update Mashups that Use RAQL Queries	108
■ Update the Snapshots Repository Connection Information	109

Read and Complete All Previous Chapters

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

Note: Presto was renamed MashZone NextGen in 9.10. This chapter will use “Presto” or “the old product” to refer to releases prior to 9.10.

Overview

This chapter explains how to complete an upgrade from Presto 3.6, 3.7, 3.8, 3.9, or 9.9 to MashZone NextGen 9.10. If you want to upgrade from releases prior to 3.6, contact Software AG Global Consulting Services.

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

Set Up the MashZone NextGen Repository

Set Up Repository When Using the Embedded Derby Database

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

1. The new MashZone NextGen comes with a prepopulated Derby repository. Go to the *new_Software AG_directory* \MashZoneNG\apache-tomee-jaxrs\bin and rename the mashzonenextgenrepository directory (for example, rename the directory mashzonenextgenrepository_orig).
2. 3.7, 3.8, 3.9, or 9.9 upgrade: Go to the *old_Software AG_directory* \Presto\apache-tomee-jaxrs\bin directory and copy the prestorepository and mashzonerepository directories to the *new_Software AG_directory* \MashZoneNG\apache-tomee-jaxrs\bin directory. Rename the prestorepository directory to mashzonenextgenrepository in the new installation.
3. 3.6 upgrade: Go to the *old_Software AG_directory* \Presto\prestorepository\derby directory and copy the prestorepository directory to the *new_Software AG_directory* \MashZoneNG\apache-tomee-jaxrs\bin directory. Rename the prestorepository directory to mashzonenextgenrepository in the new installation.

Set Up Repository When Using Any Other Database

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

1. Go to the *old_Software AG_directory \Presto\apache-tomee-jaxrs\lib* directory and copy the JDBC driver jar file for your database to the *new_Software AG_directory \MashZoneNG\apache-tomee-jaxrs\lib* directory.

Database	JDBC Driver Jar File
MySQL	mysql-connector-java-5.1.31.jar
Oracle	ojdbc6.jar
PostGres	postgresql-9.2.1004.jdbc4.jar
SQL Server	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 \MashZoneNG\apache-tomee-jaxrs\webapps\presto\WEB-INF\lib* directory and delete the *jackbe-presto-rds-postgre-derby-9.10.jar* file.
 - b. Go to the *new_Software AG_directory \MashZoneNG\prestorepository* directory and copy the *jackbe-presto-rds-oracle-mysql-mssql-9.10.jar* file to the *new_Software AG_directory \MashZoneNG\apache-tomee-jaxrs\webapps\presto\WEB-INF\lib* directory.
3. 3.7, 3.8, 3.9, or 9.9 upgrade: Go to the *old_Software AG_directory \Presto\apache-tomee-jaxrs\conf* directory and copy the *tomee.xml* file to the *new_Software AG_directory \MashZoneNG\apache-tomee-jaxrs\conf* directory.
4. 3.6 upgrade: Configure the new MashZone NextGen to use the existing repository. For instructions, go to the new MashZone NextGen Administration documentation, and then to Getting Started with MashZone NextGen, and then to Move the MashZone NextGen Repositories to a Robust Database Solution. Skip steps 1 through 4, and start with step 5.

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 documentation.
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 the MashZone NextGen Administration documentation.
 - The `userRepositoryLdap.properties` file, located in the external configuration directory.

Update When No External Configuration Directory Was Used

1. Copy any extensions you deployed from the old product to the corresponding directory for the new MashZone NextGen. See the MashZone NextGen Administration documentation for a complete list of extensions you might need to copy.
2. Manually migrate the changes you made in the old configuration files listed below from the old 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 the MashZone NextGen Administration documentation.
 - The `userRepositoryLdap.properties` file, located in the *old_Software* *AG_directory* \Presto\webapps\presto\WEB-INF\classes directory.

Copy Integrated MashZone Data and Run Upgrade Commands

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

1. MashZone NextGen's Integrated MashZone Server stores additional data. Copy the data to the new MashZone NextGen installation as follows:

- a. Go to the *old_Software AG_directory* \Presto\mashzone\data\resources directory and copy the files to the *new_Software AG_directory* \MashZoneNG\mashzone\data\resources directory.
 - b. Go to the *old_Software AG_directory* \Presto\mashzone\data\jdbcdrivers directory and copy the files to the *new_Software AG_directory* \MashZoneNG\mashzone\data\jdbcdrivers directory.
 - c. Go to the Admin Console in MashZone NextGen Hub. Expand the MashZone section, click **Server Settings**, and click the **Resources** tab. Review the paths to resource directories. If any paths point to directories in the old installation, copy those directories to the new MashZone NextGen installation and then update the paths in MashZone Administration to point to the correct location in the new MashZone NextGen installation.
2. Start the MashZone server. Open a command window or shell, go to the *new_Software AG_directory* \MashZoneNG\prestocli\bin directory, and run these commands:

```

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

```

3.9 Upgrade: Copy Integrated MashZone Data and Run Upgrade Commands

1. If you used the Integrated MashZone Feed Editor in Presto 3.9, and you uploaded files or JDBC drivers for use in the Feed Editor, you must copy the files and drivers to the new MashZone NextGen installation. In addition, the Feed Editor is now hidden by default and needs to be re-enabled.
 - a. Go to the *new_Software AG_directory* \MashZoneNG\apache-tomee-jaxrs\webapps\presto\WEB-INF\classes directory and open the presto.config file in a text editor. Find the lines shown below.

```

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

```

Set the lines to false and save the file.
 - b. Go to the *old_Software AG_directory* \Presto\mashzone\data\resources directory and copy the files to the *new_Software AG_directory* \MashZoneNG\mashzone\data\resources directory.
 - c. Go to the *old_Software AG_directory* \Presto\mashzone\data\jdbcdrivers directory and copy the files to the *new_Software AG_directory* \MashZoneNG\mashzone\data\jdbcdrivers directory.
 - d. Start the new MashZone NextGen server and go to the Admin Console in MashZone NextGen Hub. Expand the MashZone section, click **Server Settings**, and then click the **Resources** tab. If any paths point to resource directories in the

- old installation, copy those directories to the new installation and then update the paths to point to the copied directories.
2. Open a browser and enter the URL `http://MashZone NextGen_server:MashZone NextGen_port/presto/hub/dev/`.
 3. Log on as administrator. The default user name and password are Administrator \manage.
 4. Under the **Pick an API** frame in the left navigation area, expand **PolicyService** then select **reloadPermissionSets**.
 5. Click **Run** in the upper right corner. MashZone NextGen executes the **reloadPermissionSets** API call, which populates the **Request** frame.
 6. Copy RAQL user-defined functions deployed in Presto 3.9 to the new MashZone NextGen. Go to the *old_Software AG_directory* \Presto\raql-udfs directory and recursively copy all files other than the SampleRaqlLib file to the *new_Software AG_directory* \MashZoneNG\raql-udfs directory.
 7. Upgrade the JDBC configuration. For instructions, see the *MashZone NextGen User and Developer Guide*.

9.9 Upgrade: Upgrade JDBC Configuration and Run Upgrade Commands

1. Upgrade the JDBC configuration. For instructions, see the *MashZone NextGen User and Developer Guide*.
2. Open a command window or shell, go to the *new_Software AG_directory* \MashZoneNG\prestocli\bin directory, and run these commands:

```
.\padmin.bat importPolicy -u Administrator -w manage
-f ..\..\upgrade\9.10\PrestoFilesPolicy.xml
.\padmin.bat runRequestFile -u Administrator -c -w manage
-f ..\..\upgrade\9.10\upgrade_9.10.0.jump.txt
```

3.6 or 3.7 Upgrade: Update Mashups that Use RAQL Queries

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

Update the Snapshots Repository Connection Information

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

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

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

3. Update the new Snapshots configuration to point to the new Snapshots repository as follows:
 - a. Open the **JDBC Configuration** menu, click **Datasources**, click **SnapshotDatasource**, and then click **Edit**.
 - b. Edit the configuration properties as needed. For information on the properties, see the section on adding a data source in the MashZone NextGen Administration documentation.
 - c. Click **Save**.

13 Migrate the Mobile Suite

■ Read and Complete All Previous Chapters	112
■ 9.7, 9.8, or 9.9 Upgrade: Migrate Mobile Designer	112
■ Migrate Mobile Administrator	112

Read and Complete All Previous Chapters

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

9.7, 9.8, or 9.9 Upgrade: 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 remove the `MOBILE_DESIGNER` environment variable, do the following:

System	Steps
Windows	Go to the Control Panel, click System , and then click Advanced System Settings . On the Advanced tab, click Environment Variables and delete the <code>MOBILE_DESIGNER</code> environment variable. Then restart your system.
Mac OS X	Delete the <code>MOBILE_DESIGNER</code> environment variable from your <code>.bash_profile</code> , your <code>\etc\launchd.conf</code> 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.

Migrate Mobile Administrator

Overview

You will complete the Mobile Administrator upgrade by:

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

Run the Updater and Verify Configurations

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

Update Build Nodes

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

Update the App Store Client

1. Go to the https://MobileAdministrator_URL/apps/appfresh-mobile-client/build page.

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

14 Migrate OneData

■ Read and Complete All Previous Chapters	116
■ Before Migrating	116
■ Run the OneData Migration Utility	116
■ Complete the OneData Upgrade	116

Read and Complete All Previous Chapters

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

Before Migrating

1. If the new and old OneDatas are on the same machine, make sure the old OneData is shut down.
2. Go to the *new_Software AG_directory* \profiles\ODE\bin\migrate directory, open the migrate.bat and migrate.sh files, and set the JAVA_DIR property to *new_Software AG_directory* \jvm\jvm.

Run the OneData Migration Utility

The OneData migration utility migrates configuration data.

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

For a general description of migration utility behavior, see "[Migration Utilities](#)" on page 39.

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

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

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

Complete the OneData Upgrade

1. If you customized the old OneData Java Service Wrapper, copy the customizations from the old wrapper_conf or custom_wrapper.conf files to the new custom_wrapper.conf file. The files are located in the *Software AG_directory* \profiles\ODE\configuration directory.

2. If you are using JMS with OneData, copy the client jar files you backed up earlier to the *new_Software AG_directory* \profiles\ODE\webapp\onedata\WEB-INF\lib directory.
3. If you are using Kerberos authentication with OneData, reconfigure it using the backups of the web.xml, server.xml, jaas.config, custom_wrapper.conf and com.softwareag.jaas.realm.pid-SSO_realm_name.properties files you created earlier. For instructions, see *Administering webMethods OneData*.
4. Go to the *new_Software AG_directory* \profiles\ODE\configuration \com.softwareag.platform.config.propsloader directory. Open the three com.softwareag.catalina.resource.pid-onedataschema Db1.properties files, where *schema* is md, wa, or ra. Replace values for database parameters such as maxIdle, maxActive and maxWait with the values you noted earlier from the old installation.

15 Migrate Software AG Event Routing and Perform Final Upgrade Tasks for All Products

■ Migrate Software AG Event Routing	120
■ Complete Final Upgrade Tasks for All Products	121

Migrate Software AG Event Routing

Event Routing components are automatically installed with products. These components are Event Type Store, Event Persistence, Event Routing, and NERV.

Before Migrating

Make sure none of the new products are running.

Run the Software AG Event Routing Migration Utility

The Software AG Event Routing migration utility migrates the following:

- Event Type Store.
- 9.9 upgrade: Event Routing configurations for all products.
- Software AG Designer Event Bus Console connection configurations that use NERV.
- NERV configuration bundles, if you deployed any.

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

For a general description of migration utility behavior, see "[Migration Utilities](#)" on page 39.

On the machine that hosts the new products, open a command window or shell, go to the *new_Software AG_directory* \common\migrate\EDA\bin directory, and run the command below. For the *migrateold_release* sbs.dat file, specify the source release number without periods (for example, 980).

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

Migrate Undelivered Events

Go to the *new_Software AG_directory* \install\log directory and open the migrationLog.txt file. If you see the message below for one or more products, events that were delivered to those products in the old installation were not acknowledged before the products were shut down.

```
WARNING: There are still events left in IP storage location_of_old_storage_file.
These events will not be migrated automatically and will not be redelivered by
Event Routing in the new installation. For information how to migrate these
events manually, see the upgrade guide.
```

To enable Event Routing to deliver the events to products in the new installation, you must copy the old storage files to the new installation. For each message, do the following:

- If the message provides the full path to the old storage file, the old storage file is in the default location (that is, the *new_Software AG_directory* \profiles \product \configuration \event \routing \runtime \storage directory). Copy the old storage file to the same location in the new installation directory.
- If the message identifies the old storage file (for example, c:\EventRoutingStorage \SomeName_Reliable.mem) but does not provide the full path, the old storage file is in a non-default location.
 1. In the migrationLog.txt file, determine to which product the message applies.
 2. Go to the *new_Software AG_directory* \profiles \product \configuration \event \routing \runtime directory.
 3. Open the configuration.json file and find the storage-location property.
 4. Copy the old storage file to the directory specified on the property.

9.9 Upgrade: Migrate Event Persistence Services

If you used Event Persistence to persist events in the old installation, you must manually migrate the old product configurations for the Event Persistence services to the new installation. For each product in the new installation, go to the *old_Software AG_directory* \profiles \product \configuration \event \routing \services directory and copy the entire EventPersistence directory (including all service type configurations) to the *new_Software AG_directory* \profiles \product \configuration \event \routing \services directory.

Update Paths

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

Complete Final Upgrade Tasks for All Products

- Before performing any task in this chapter, read all chapters that precede this chapter carefully and complete all tasks that apply to your products in the order in which they are presented in those chapters.
- Configure your new products. For instructions, see the product documentation for the new release.

- If you installed your new products on a different machine than your old products, make sure to update host names in your new products, in the connections between your products, and in your database tables. This guide indicates many locations in which to update host names, but make sure you also specify the correct host names when you configure your new products using the instructions in the product documentation. Also make sure any absolute paths in the new configuration files point to valid locations, or change them to be correct for the new machine. If the machine has a different operating system or hardware, make sure your JVM settings are correct.
- Read the product readmes for the new release, including the readme for the Software AG Infrastructure. All readmes are available on the Software AG Documentation website. The product readmes contain this information:
 - Critical information and known and resolved issues for your products.
 - Changes (that is, additions, changes, deprecations, and removals) to product behavior, services, parameters, properties, and APIs. This information is especially important because you might need to modify product files or assets after migration to accommodate the changes.

Product readmes contain information for the 9.5 release and later. Read the information for your old release + 1 through the new release. For example, if you are upgrading from 9.0 to 9.9, read the information for releases 9.5 through 9.9.

- After installation, you might have set Windows services for products to Manual, and disabled scripts that start UNIX daemons, to avoid automatically starting both old and new products. When your new environment is ready, after you stop running the old products and when you want to start running the new ones, you can reset the Windows services to Automatic, and re-enable the UNIX scripts.