

Administering My webMethods Server

Version 10.11

October 2021

This document applies to My webMethods Server 10.11 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 © 2004-2024 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

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

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

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

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

Document ID: MWS-AG-1011-20240129

Table of Contents

About This Guide.....	9
Exporting My webMethods Server Assets.....	10
Document Conventions.....	10
Online Information and Support.....	11
Data Protection.....	12
Deprecation of webMethods Broker.....	12
I Startup and Configuration.....	13
1 Getting Started with My webMethods Server.....	15
Introduction to My webMethods Server.....	16
Starting and Stopping My webMethods Server on Windows.....	17
Automatically Starting My webMethods Server on UNIX.....	17
Logging Into My webMethods Server.....	17
Logging Out of My webMethods Server.....	18
Enabling Access for Administrative Accounts.....	18
2 Changing the My webMethods Server Configuration.....	21
Post-Installation Configuration Changes.....	22
Using My webMethods Server with a MySQL Database.....	22
Did You Install Without Specifying a Database?.....	23
Guidelines for Multiple My webMethods Server Instances.....	23
Changing the HTTP Listener Port for a Standalone Server.....	24
Changing HTTP and HTTPS Listener Ports.....	24
Changing the JMX Listener Port.....	25
Changing the JCR Repository RMI Listener Ports.....	26
Specifying the Java Location for My webMethods Server.....	26
Managing Redirection in My webMethods Server.....	28
3 My webMethods Server and HTTPS.....	31
Using My webMethods Server as an HTTPS Client.....	32
Communicating with webMethods Applications Using HTTPS.....	35
Managing Authentication Certificates as My webMethods Administrator.....	36
4 Using My webMethods Server with Web Servers.....	43
Integration with Web Servers.....	44
My webMethods Server and Apache.....	44
My webMethods Server and IIS.....	44
5 Using My webMethods Server with Docker.....	45
About My webMethods Server and Docker Containers.....	46
Running My webMethods Server in Docker Containers.....	46
Considerations when Working with My webMethods Server Containers.....	46
Creating and Customizing Docker Images for My webMethods Server Installations...47	
Running and Customizing My webMethods Server Containers.....	50
About the Optimized My webMethods Server Container Startup.....	61
Using My webMethods Server Containers with webMethods Microservices Runtime.62	
Monitoring My webMethods Server Containers.....	63
Clustering My webMethods Server Containers.....	68
6 Running My webMethods Server from the Command Line.....	71

Basic Command Line Syntax for My webMethods Server.....	72
Executing My webMethods Server Commands.....	73
Accessing the Command Line Help Contents.....	73
My webMethods Server Instance Administration Commands.....	74
My webMethods Server Instance Operation Commands.....	79
My webMethods Server Instance Configuration Commands.....	82
My webMethods Server Service Management Commands.....	84
My webMethods Server OSGi Profile Commands.....	86
Start, Stop and Execute My webMethods Server Commands on Multiple Server Instances.....	87
Configuring JVM Properties for My webMethods Server Commands.....	88
Log Files for mws Commands.....	88
7 Modifying Configuration Files.....	91
The Java Service Wrapper.....	92
Configuring JVM Settings for My webMethods Server.....	95
Modifying Configuration Files Stored in the Database.....	100
Configuring My webMethods Server to Run in 32-bit on Solaris, HP-UX, or Linux...	102
Configuring HTTP Listeners to Use a Single IP Address.....	103
Configuring Whether Diagnostics Are Executed at Startup.....	105
II My webMethods Administrator Functions.....	107
8 Managing My webMethods Configuration.....	109
Managing Directory Services.....	110
Managing External Directory Services.....	112
Managing External Data Sources.....	129
Managing Email Settings.....	137
My webMethods Server and Multi-Factor Authentication.....	138
Managing Calendars.....	140
9 Searches for Users, Groups, and Roles.....	141
Searching for Existing Users, Groups, or Roles.....	142
Advanced Searches.....	142
Working with Saved Searches.....	144
Exporting Search Results to a .csv File.....	146
10 Managing Users and Groups.....	149
About Managing Users and Groups.....	150
Managing Users.....	151
Managing User Data.....	161
Managing Groups.....	163
11 Managing Permissions.....	169
Managing Permissions in My webMethods.....	170
Managing Access Privileges and Functional Privileges.....	173
Managing Permissions for an Individual Resource.....	174
Using Security Realms.....	176
12 Managing Roles and Access to My webMethods.....	183
About Roles in My webMethods Server.....	184
Granting Users Access to My webMethods and the My webMethods Users Role.....	185
Creating Roles.....	186
Editing Information for a Role.....	192
Deleting Roles.....	193

Defining Dynamic Attributes Associated with a Role.....	194
13 My webMethods Server Clustering.....	199
How a My webMethods Server Cluster Works.....	200
Planning Your My webMethods Server Cluster.....	204
Considerations When Building a My webMethods Server Cluster.....	206
Modifying the Cluster Configuration.....	207
Modifying Resource Locking Settings for a Cluster.....	217
Configuring the Number of Purge Threads for a Cluster.....	218
Monitoring and Controlling Your Cluster.....	218
Removing a Component from a Cluster.....	222
Working with the cluster.xml File.....	222
Creating a Cluster Node from an Image.....	225
Partitioning Applications on Cluster Nodes.....	227
III System Administrator Functions.....	239
14 Attribute Providers.....	241
What are Attribute Providers?.....	242
Using Attribute Providers.....	243
Managing the Display of Principal Attribute Providers.....	247
15 Managing Security.....	251
About My webMethods Server Security.....	252
Managing Authentication.....	263
Configuring Kerberos Authentication.....	266
Configuring NTLM Authentication.....	271
Configuring NTLMv2 Authentication.....	272
Configuring External Configuration Credentials.....	274
Configuring My webMethods Server Single Sign-On.....	277
Configuring OAuth 2.0 Authentication.....	285
Clearing Session Passwords from Memory.....	289
Retaining Session Passwords in Memory.....	289
Turning On or Off Auto Complete for Usernames and Passwords.....	290
Controlling the Number of Failed Login Attempts.....	290
Controlling Login IP Ranges.....	291
Encrypting Passwords for Global Environment Variables.....	293
Allowing Context Impersonation.....	294
Using Password Complexity Policies.....	294
Working with Response Header Rules.....	299
16 Analysis, Reporting, and Troubleshooting.....	305
About Analysis, Reporting, and Troubleshooting.....	306
Controlling Server Logging.....	306
Viewing Logging Messages.....	318
Managing Security Audit Logging.....	318
Monitoring Real-Time User Activity.....	319
Collecting Data About Server Events.....	319
Collecting Data About Database Changes.....	322
My webMethods Server Diagnostic Tools.....	322
17 My webMethods Server Configuration.....	323
About My webMethods Server Configuration.....	324
Managing Aliases.....	324

Deploying My webMethods Server Components.....	328
Configuring My webMethods Server Cache.....	331
Displaying System Information.....	337
18 Managing My webMethods Server Content.....	339
Migrating My webMethods Server Content.....	340
Managing Content Storage.....	340
Publishing Portlets as an Administrator.....	343
Rebuilding the Search Index.....	344
Adding Custom JAR Files.....	345
19 Managing the User Interface.....	347
Locale Administration.....	348
What Are Server Rules?.....	348
Creating Locale Rules.....	349
Creating Login Page Rules.....	351
Creating Start Page Rules.....	354
Creating Rendering Rules.....	356
Modifying a Rule.....	358
Copying a Rule.....	360
Managing the Evaluation Order for Rules.....	360
Removing a Rule.....	361
Managing Skin Rules.....	361
Managing Shell Rules.....	364
20 Working with the Common Directory Services API.....	369
Managing User Information with the Common Directory Service API.....	370
About the Common Directory Services API.....	371
Prerequisites.....	371
About CDS Version Interoperability.....	372
CDS Code Examples.....	373
21 Sending Mobile Notifications from My webMethods Server.....	375
Sending Push Notifications to Mobile Devices.....	376
Configuring Push Notifications in My webMethods Server.....	377
IV Server Page Development.....	379
22 Managing Pages in My webMethods Server.....	381
About Page Development.....	382
About Custom Folders and Pages.....	382
About Customizing the My webMethods Navigation.....	399
About Customizing the My webMethods Look-And-Feel.....	404
Building a Simple Front-End Page to My webMethods.....	408
Creating Links for Single Sign-On.....	409
23 Managing Workspaces in My webMethods Server.....	411
About Workspaces.....	412
Administration Tasks for Workspaces.....	412
Expert User Features for Workspace Development.....	416
Workspace Actions You Can Perform from the Workspace Management Page.....	422
About the My webMethods Tools Navigation.....	429
24 Customizing Skins.....	433
What Are Skins?.....	434
Creating and Modifying a New Skin.....	436

Using the Skin Administration Page.....	439
Make-up of a Skin Package.....	448
25 Working with Shells in My webMethods Server.....	459
What Are Shells?.....	460
Creating a New Shell.....	460
Modifying a Shell.....	461
Inserting Extra Tags into the HTML <head> Element.....	462
Using an Alias with a Shell Section.....	462
Deleting a Shell.....	463
Making an Empty Shell Section.....	463
V Using Command Central to Manage My webMethods Server.....	465
26 Administering My webMethods Server.....	467
Administering My webMethods Server Instances.....	468
Configuring My webMethods Server Ports.....	471
Configuring Directory Services.....	472
Configuring My webMethods Server Email.....	479
Working with My webMethods Server Environment Variables.....	480
Monitoring KPIs of My webMethods Server Instances.....	481
Using Trusted Authentication to Connect to My webMethods Server.....	482
27 Using the Command Line to Manage My webMethods Server.....	483
Commands that My webMethods Server Supports.....	484
Configuration Types that My webMethods Server-ENGINE Supports.....	486
Lifecycle Actions for My webMethods Server-ENGINE.....	488
My webMethods Server Instance Management.....	489
Run-time Monitoring Statuses for My webMethods Server-ENGINE.....	493
Run-time Monitoring States for My webMethods Server.....	494
28 Authenticating My webMethods Server.....	497
Changing the Authentication Mode for My webMethods Server.....	498
Enabling Access for Administrative User Accounts for My webMethods Server in Command Central.....	498
Using Unix Shell Scripts to Change Connection Credentials for Managed Products.....	499
Verifying the Outbound Authentication Settings.....	499
Accessing Administrative Interfaces Through Command Central.....	499

About This Guide

- [Exporting My webMethods Server Assets](#) 10
- [Document Conventions](#) 10
- [Online Information and Support](#) 11
- [Data Protection](#) 12
- [Deprecation of webMethods Broker](#) 12

This guide explains how to configure and manage My webMethods Server as a system administrator. The guide explains how, as a My webMethods Administrator (not the same as a system administrator), you can manage users, groups, and roles for the applications that run on My webMethods Server. In addition, the guide describes how to develop and manage pages for display by My webMethods Server.

Exporting My webMethods Server Assets

For information on extracting My webMethods Server assets for use with Deployer, see *webMethods Deployer User's Guide*.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Software AG Documentation Website

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

Software AG Empower Product Support Website

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

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

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

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

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

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

Software AG Tech Community

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

- Access product documentation, if you have Tech Community 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.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/u/softwareag> and discover additional Software AG resources.

Data Protection

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

Deprecation of webMethods Broker

webMethods Broker is deprecated for use with webMethods 10.2. If you are starting development using webMethods 10.2, you should use Software AG Universal Messaging instead of webMethods Broker. If you are upgrading to webMethods 10.2, you should consider migrating to Universal Messaging. If you choose to continue to use webMethods Broker, you will still be fully supported, but only until the announced end-of-life dates for webMethods Broker. For details, see <https://empower.softwareag.com/brokerendoflife/>.

I Startup and Configuration

1	Getting Started with My webMethods Server	15
2	Changing the My webMethods Server Configuration	21
3	My webMethods Server and HTTPS	31
4	Using My webMethods Server with Web Servers	43
5	Using My webMethods Server with Docker	45
6	Running My webMethods Server from the Command Line	71
7	Modifying Configuration Files	91

1 Getting Started with My webMethods Server

■ Introduction to My webMethods Server	16
■ Starting and Stopping My webMethods Server on Windows	17
■ Automatically Starting My webMethods Server on UNIX	17
■ Logging Into My webMethods Server	17
■ Logging Out of My webMethods Server	18
■ Enabling Access for Administrative Accounts	18

Introduction to My webMethods Server

My webMethods Server is a run-time container for functions made available by webMethods applications. The user interface in which you perform these functions is called My webMethods. My webMethods provides a ready-made environment in which users can perform functions on webMethods applications, and administrators can manage access to those functions. In addition, My webMethods Server gives you the capability to develop additional user interface pages, and a broad-based set of administrative tools with which to manage the increased capabilities.

My webMethods Server recognizes two types of administrators, based on the functions they perform:

- **System administrator** - The system administrator for My webMethods Server. This user can manage My webMethods Server, including analysis, configuration, content, and user management. The user ID is `SysAdmin` and the password is specified during installation. Depending on the options you select during installation, My webMethods Server might prompt the `SysAdmin` user to specify a new password on the first login. This user does not use the My webMethods user interface.
- **My webMethods Administrator** - The default administrator of My webMethods. This user can perform user management functions and manage external directory services. The user ID is `Administrator`. By default, this user is disabled after installation. The `SysAdmin` user must log in and enable the `Administrator` account.

In some cases, both types of administrator can perform the same functions, such as performing user management. Where there are differences between the My webMethods and system user interfaces, the procedures describe both.

This guide is organized into multiple parts. The following table lists the major parts of the guide and the activities that each part covers.

Activity	More Information
Getting started as an administrator, changing the configuration of My webMethods Server, using external web servers, and running servers from the command line.	Startup and Configuration
Managing users, groups, and roles as a My webMethods Administrator.	<i>My webMethods Administrator Functions</i>
Managing the advanced capabilities of My webMethods Server as a system administrator.	<i>System Administrator Functions</i>
Developing and managing user interface pages.	Server Page Development
Administering My webMethods Server with Command Central.	<i>Using Command Central to manage My webMethods Server</i>

In addition, you can find more information about My webMethods Server in the following guides:

- Installation and initial configuration of My webMethods Server - *Installing Software AG Products*.
- The basic activities an individual user can perform on My webMethods applications - *Working with My webMethods*.

Starting and Stopping My webMethods Server on Windows

You can start My webMethods Server by starting the Windows service for My webMethods Server.

To shut down the server, stop the Windows service for My webMethods Server.

The name of the service is Software AG My webMethods Server 10.2. When more than one instance of the service exists, the second instance has an index value of (2), the third instance has an index value of (3), and so forth.

Automatically Starting My webMethods Server on UNIX

If you installed My webMethods Server on a UNIX system and want My webMethods Server to start automatically each time you start your system, execute the My webMethods Server service registration script, as follows:

```
Software AG_directory/MWS/bin su -c ./mws.sh  
-s serverInstance installservice serverUserAccount
```

For more information about My webMethods Server startup, including optional parameters, see [“Running My webMethods Server from the Command Line” on page 71](#).

Logging Into My webMethods Server

My webMethods Server has administration interfaces that you access using a web browser. Only the SysAdmin user can log into My webMethods Server immediately after installing and starting the server. All other default administrative accounts are disabled, and must be enabled by the SysAdmin user in order to access My webMethods.

➤ To log into My webMethods Server as SysAdmin

1. Access the My webMethods Server Login page by entering a URL in a web browser:

```
http://host:port
```

where:

- *host* is the host name of the machine on which My webMethods Server is installed.

- *port* is the port on which My webMethods Server listens for incoming requests. The default port for My webMethods Server is 8585.

For example, if the host name is `rubicon.company.com` and it uses the default port (8585), type the following URL:

```
http://rubicon.company.com:8585
```

2. In the **Username** field, enter `SysAdmin`, and for **Password**, specify the password, configured during installation.

Depending on the options, selected during installation My webMethods Server might require changing the password of the `SysAdmin` user on the first login. If prompted, specify a new password for this account on the My webMethods Server login screen, and confirm the new password.

3. Click **Log In**.

After you log in, My webMethods Server displays the Administration Dashboard.

Logging Out of My webMethods Server

Perform the following procedure to log out of My webMethods Server.

➤ To log out of My webMethods Server

- Click the **Logout** link, which is located at the top of all My webMethods pages.

Enabling Access for Administrative Accounts

By default all user accounts with administrative privileges, except the `SysAdmin` user are disabled after installation. The `SysAdmin` user must explicitly enable, and provide new passwords for such accounts. After the `SysAdmin` user enables the user accounts, administrative users can change their passwords on the **User Information** panel of the **My Profile** page in My webMethods.

➤ To enable access and change passwords for administrative accounts

1. As system administrator, go to **Administration Dashboard > User Management > Manage Users**.
2. Search for the user you want to edit. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).
3. In the search results, click any link in the row of the user you want to edit or click .
4. Deselect the **Login Disabled** checkbox.

5. On the **User Information** panel, type a new password in the **Password** field.
6. In the **Confirm Password** field, type the new password again for confirmation.
7. Click **Apply**.

2 Changing the My webMethods Server Configuration

- Post-Installation Configuration Changes 22
- Using My webMethods Server with a MySQL Database 22
- Did You Install Without Specifying a Database? 23
- Guidelines for Multiple My webMethods Server Instances 23
- Changing the HTTP Listener Port for a Standalone Server 24
- Changing HTTP and HTTPS Listener Ports 24
- Changing the JMX Listener Port 25
- Changing the JCR Repository RMI Listener Ports 26
- Specifying the Java Location for My webMethods Server 26
- Managing Redirection in My webMethods Server 28

Post-Installation Configuration Changes

When you install My webMethods Server, it has a default configuration. You can specify the type and location of the database used by the server and the HTTP port the server uses, but nothing else. After the installation is completed and you have a running instance of My webMethods Server, you can make changes to the configuration.

Using My webMethods Server with a MySQL Database

To use My webMethods Server with MySQL, you must configure the mode of the MySQL database before starting My webMethods Server for the first time. The MySQL modes that My webMethods Server requires are `ANSI_QUOTES` and `PIPES_AS_CONCAT`.

When using My webMethods Server with a MySQL database, deploying `.war` files might fail. If such an issue occurs, increase the values of the following parameters in the MySQL configuration file, and restart the MySQL database server.

```
innodb_buffer_pool_size = 5G
innodb_log_file_size = 80M
max_allowed_packet = 24M
```

For more information about configuring MySQL, refer to the MySQL documentation.

Configuring My webMethods Server to Use MySQL Community Edition

MySQL Community Edition requires the use of a native driver. Before you can use My webMethods Server with MySQL Community edition, you must download the driver and add it as a bundle in the My webMethods Server installation.

➤ To configure My webMethods Server for MySQL Community Edition

1. Download the MySQL native driver `mysql-connector-java-version-bin.jar` from the following location:

<https://dev.mysql.com/downloads/connector/j/5.1.html>

2. Copy the `.jar` file to the `Software AG_directory \MWS\lib` directory.
3. Create a `driver-name.bnd` text file in the `Software AG_directory \MWS\lib` directory, where `driver-name` is the name of the `.jar` file.

For example, `mysql-connector-java-version.bnd`.

4. Provide instructions for the OSGi bundle conversion in the `.bnd` text file, by replacing the values in *italics* in the following example.

```
# attach as fragment to the caf.server bundle
Fragment-Host: com.webmethods.caf.server
Bundle-SymbolicName: mysql-connector-java
Bundle-Version: 5.1.41
Include-Resource: mysql-connector-java-5.1.41.jar
-exportcontents: *
Bundle-ClassPath: mysql-connector-java-5.1.41.jar
Import-Package: *;resolution:=optional
```

For *Bundle-Version*, specify the version number of the .jar file, or any unique number.

- At a command line prompt, move to the bin directory of server:

```
Software AG_directory\MWS\bin
```

- Update the server instance using the following command:

```
mws.bat -s serverName update
```

- Start My webMethods Server.

Did You Install Without Specifying a Database?

It is possible to install My webMethods Server without specifying a database. In this case, if you want to use an external database, you need to configure a database connection for My webMethods Server before the server will start. To set or modify external database connections, use the `mws.db.xml` file, which you can find at this location:

```
Software AG_directory \MWS\server\serverName\config
```

You need to specify the following values:

The following table lists the values that you must specify in the `mws.db.xml` file to configure the database connection.

Element	Value
<URL>	The URL to the database server
<USER>	The database user name
<PASSWORD>	The password of the database user

After you save and close the `ms.db.xml` file, start My webMethods Server.

Guidelines for Multiple My webMethods Server Instances

You can run multiple instances of My webMethods Server on the same machine, but each server instance must have its own external resources. Running multiple server instances on the same

machine is not the same as clustering. For more information about My webMethods Server clustering, see “[My webMethods Server Clustering](#)” on page 199. The following guidelines apply to running two or more server instances on the same machine:

- Each My webMethods Server instance must have its own database; for a given database server, the following configuration entries must be unique among all My webMethods Server instances that use the same database server:
 - Database user name
 - Database name or tablespace name
- For My webMethods Server instances running concurrently on the same machine, the following host/port number combinations, if used, must be unique among all servers:
 - HTTP port. The default is 8585.
 - HTTPS port (if used).
 - Java Management Extensions (JMX) port. The default is 5002.

Changing the HTTP Listener Port for a Standalone Server

Note:

If you are using webMethods Monitor and change the My webMethods Server default listening port of 8585, you get an Access Denied error on the Administration > Business Processes page in My webMethods. To resolve this issue, specify the non-default port number in the **MWS SAML Resolver URL** field on the Settings > Resources page in Integration Server Administrator.

➤ **To change only the HTTP listener port for a standalone server instance**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Basic Configuration.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Basic Configuration.**
2. In the **HTTP PORT** field, type the new port number and click **Submit**.
3. Restart My webMethods Server.

Changing HTTP and HTTPS Listener Ports

Note:

If you are using webMethods Monitor and change the My webMethods Server default listening port of 8585, you get an Access Denied error on the Administration > Business Processes page in My webMethods. To resolve this issue, specify the non-default port number in the **MWS SAML Resolver URL** field on the Settings > Resources page in Integration Server Administrator.

➤ To change HTTP and HTTPS listener ports for a My webMethods Server instance

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Advanced or Cluster Configuration.**
 - As system administrator, click **Administration Dashboard > Configuration > Cluster Administration > Advanced or Cluster Configuration.**
2. In the **HTTP PORT** field, type the port number to be used by the HTTP listener.
This field must always have a valid port number.
3. In the **HTTPS PORT** field, type the port number to be used by the HTTPS listener.
A value of 0 (zero) in this field disables the listener.
For information about the default digital certificates, used by the HTTPS Listener, see [“Certificates Used for Secure Connections” on page 34.](#)
4. Click **Submit**.
5. Restart My webMethods Server.

Changing the JMX Listener Port

The default Java Management Extensions (JMX) listener port is 5002. If multiple instances of My webMethods Server are running on the same computer, or if another application is already using that port, you must assign a different port number. You change the JMX listener port in the `com.softwareag.jmx.connector.pid-5002.properties` file.

➤ To change the JMX listener port

1. In a text editor, open the `com.softwareag.jmx.connector.pid-5002.properties` file for the server instance. The file is in the `Software AG_directory \profiles\MWS_serverName\configuration\com.softwareag.platform.config.propsloader` directory.
2. Edit the `port=number` line with the new JMX listener port number, and then save the file.
3. Restart My webMethods Server.

Changing the JCR Repository RMI Listener Ports

When using a remote Java Content Repository (JCR), the connection to the JCR is implemented using an RMI to communicate with the client My webMethods Server. My webMethods Server uses the following ports for the JCR RMI connection:

- **rmiPort.** The port for the RMI registry. The default port number is 10999.
- **rmiServerObjectPort.** The port used by the remote client to communicate with the remote object that is provided by My webMethods Server. The default port number is 10998. A value of 0 (zero) generates a random port number.

Important:

If your server is protected by a firewall and you want to allow remote JCR connections to My webMethods Server, you must configure your firewall to open both the **rmiPort** and the **rmiServerObjectPort**. In that scenario, you should use a non-zero value for the **rmiServerObjectPort** so that the port number does not change every time My webMethods Server is restarted.

➤ To change the **rmiPort** or **rmiServerObjectPort** number from the default

1. As system administrator, go to **Administrative Folders > Administration Dashboard > Configuration > CAF Application Runtime Configuration**.
2. In the **Keywords** field, type `wm_mws_config` to search for the `wm_mws_config` deployed application.

My webMethods Server returns the Administration application in the search results.
3. Click **Administration > Web Application > Environment Entries**.
4. In the **rmiPort** and **rmiServerObjectPort** fields, type new values for the port numbers and click **Apply**.

Specifying the Java Location for My webMethods Server

My webMethods Server must point to a Java location. By default, My webMethods Server points to the location of the JDK installed in the *Software AG_directory* /jvm directory.

You can specify a non-default JDK or JRE to be used by My webMethods Server. If you do so, do not delete the default JDK because it is used by the Software AG Installer.

Important:

If you specify a non-default JRE or JDK, apply maintenance updates from the appropriate vendor on a regular basis, as you would for JREs and JDKs you install yourself.

Note:

Software AG tests products only with the JDKs installed by the Software AG Installer. If you redirect products to use a different JDK or JRE and encounter issues, Software AG might require you to reproduce the issues with the JDK that is installed by the Software AG Installer.

To specify a non-default JDK or JRE, you must make changes in multiple locations.

Setenv File under /MWS/bin/

You must modify the Java location specified in the setenv.bat or .sh file located here:

Software AG_directory /MWS/bin/setenv.[bat | sh]

Open the setenv.bat or setenv.sh file in a text editor. Edit the JAVA_HOME parameter to point to the non-default JDK or JRE installation directory, then save and close the file. For example:

```
set JAVA_HOME= C:\myjava17
```

Setenv File under /profiles

You must modify the Java location specified in the setenv.bat or .sh file located here:

Software AG_directory /profiles/MWS_serverName/bin/setenv.[bat | sh]

Open the setenv.bat or setenv.sh file in a text editor. Edit the JAVA_EXEC parameter to point to the non-default JDK or JRE installation directory, then save and close the file. For example:

```
set JAVA_EXEC= "C:\myjava17\bin\java"
```

The wrapper.conf File

You must modify the Java location specified in the wrapper.conf file located here:

Software AG_directory /profiles/MWS_serverName/configuration/wrapper.conf

Open the wrapper.conf file in a text editor. Edit the wrapper.java.command parameter to point to the non-default JDK or JRE installation directory, then save and close the file. For example:

```
wrapper.java.command=C:\myjava17\bin\java
```

The custom_wrapper.conf File

You must modify the Java location specified in the custom_wrapper.conf file located here:

Software AG_directory /profiles/MWS_serverName/configuration/custom_wrapper.conf

Open the custom_wrapper.conf file in a text editor. Edit the set.JAVA_HOME parameter to point to the non-default JDK or JRE installation directory, then save and close the file. For example:

```
set.JAVA_HOME=C:\\ myjava17
```

Managing Redirection in My webMethods Server

In My webMethods Server, when you click a button or other action on a page, it can result in a redirection that takes you to an external site. While redirection is a useful feature, it has the potential to point a user to a malicious external site. Direct links created using the Link control or the Bookmark workspace tool are not affected by this issue.

A *whitelist* is a list of trusted entities, in this case trusted servers. Using the Redirection Whitelist Administration page, you can create a whitelist containing servers to which My webMethods Server can safely redirect a request.

By default, My webMethods Server does not allow redirection to any external site, making it less prone to malicious exploitation. The server does allow redirection to the cluster front end, `localhost`, and the loopback address (`127.0.0.1`), which are required for operation.

If you need to allow redirection to external servers, you can add them using the Redirection Whitelist Administration page. Servers are identified by host name or IP address. The whitelist is stored in the server database, making it available to all servers in a My webMethods Server cluster.

Adding Servers to a Whitelist

You can allow redirection to an external address by adding it to the whitelist. By default, the cluster front end, `localhost`, and the loopback address (`127.0.0.1`), are included in the list.

➤ To add an external address to the whitelist

1. As system administrator, click **Administration Dashboard > Configuration > Redirection Whitelist Administration**.
2. Click **Add Server**, type in the hostname or IP address of a trusted server, click **Apply**, and then click **Save**.

My webMethods Server will now allow URL redirection to the new server.

Removing Servers from a Whitelist

If an external address is included in the whitelist, you can remove it. By default, the cluster front end, `localhost`, and the loopback address (`127.0.0.1`), are included in the list and cannot be removed.

➤ To remove an external address from the whitelist

1. As system administrator, click **Administration Dashboard > Configuration > Redirection Whitelist Administration**.
2. Click the check box for the server to be removed, click **Remove Server**, and click **Save**.

The server is now removed from the whitelist.

3 My webMethods Server and HTTPS

- Using My webMethods Server as an HTTPS Client 32
- Communicating with webMethods Applications Using HTTPS 35
- Managing Authentication Certificates as My webMethods Administrator 36

Using My webMethods Server as an HTTPS Client

If you want to connect My webMethods Server to a server using HTTPS, you must set up the trusted CA store file (the cacerts file) that the JVM running in the My webMethods Server machine uses. The trusted CA store file must contain the CA certificates of the servers to which My webMethods Server will be issuing HTTPS requests.

For example, if you use webMethods Monitor or the WmTaskClient Package, you must identify the Integration Server to which My webMethods Server issues requests on behalf of Monitor pages. If you choose to have the requests issued using HTTPS, you must then set up the trusted CA store file of the JVM running in the My webMethods Server machine to contain the CA certificate of the Certificate Authority that signed the Integration Server certificate.

You import CA certificates into the trusted CA store file of the JVM using the JVM's keytool command. By default, the trusted CA store file is located in the following location:

```
Software AG_directory \jvm\operating_system\jre\lib\security\cacerts
```

For example, if you use Windows, the location of the trusted CA store file is *Software AG_directory \jvm\jvm\jre\lib\security\cacerts*.

Importing CA Certificates

➤ To import CA certificates into the trusted CA store file of the My webMethods Server JVM

1. Locate the CA certificate you need to add to the trusted CA store file and ensure it is available on the machine running My webMethods Server.
2. At a command line prompt, type the following command to move to the `jvm\lib\security` directory:

```
cd Software AG_directory\jvm\operating_system\jre\lib\security
```

3. Type the following command to import the CA certificate into the trusted CA store file:

```
..\..\..\bin\keytool -import -v -keystore  
cacerts -file <cacert.der> -alias <aliasName>
```

where:

- `-file <cacert.der>` identifies the path and file name of the file that contains the CA certificate you want to import
- `-alias <aliasName>` assigns an alias to the certificate to identify the entry in the key store file. Select a value that is meaningful to you.

For example, to import the CA certificate named `serverCAcert.der`, which is stored in the same directory as the `cacerts` file, and identify the new entry in the key store file as `SERVERCA`, you would use the following command:

```
..\..\..\bin\keytool -import -v -keystore
cacerts -file serverCAcert.der -alias SERVERCA
```

4. After entering the keytool command, the command prompts you for the password for the cacerts file. Type the password. By default, the password is `changeit`.
5. After entering the password, the keytool command prompts to verify that you want to import the CA certificate. Type `y` for yes.
6. To ensure that the CA certificate was successfully imported into the trusted CA store file, enter the following command:

```
..\..\..\bin\keytool -list -keystore cacerts
```

The keytool command prompts for the password for the cacerts file. Type the password.

Example

Assume that you want the WmTaskClient Package to communicate with My webMethods Server on the same computer using SSL. In this example, we use the default My webMethods Server truststore.

1. If you have not already done so, configure My webMethods Server to use an HTTPS port. For example, set the HTTPS port to 8586. For more information on how to set an HTTPS port, see [“Communicating with webMethods Applications Using HTTPS” on page 35](#).
2. In Integration Server, configure the WmTaskClient Package to communicate using the HTTPS port configured in the previous step (8586).
3. Create a temporary directory in which to store the CA certificate, such as `C:\temp`.
4. At a command line prompt, move to the directory of the JVM keytool command:

```
cd Software AG_directory\jvm\operating_system\jre\lib\security
```

5. Type the following command to extract the CA certificate from the default My webMethods Server truststore:

```
keytool -export -alias "softwareag demo" -file c:\temp\sagca.crt
-keystore Software AG_directory\MWS\server\default\config\
security\sagdemoca.jks
```

6. At the prompt, type the truststore password. For the default My webMethods Server truststore, the password is `manage`.
7. To import the CA certificate into the trusted CA store of the JVM, type the following command:

```
keytool -import -trustcacerts -file c:\temp\sagca.crt
-alias "softwareag demo" -keystore Software AG_directory\jvm\
jvm160_32\jre\lib\security\cacerts
```

Note:

If you are running on a 64-bit operating system, change `jvm160_32` to `jvm160_64`.

- At the prompt, type the password for the trusted CA store file of the JVM. By default, the password is `changeit`.
- To verify that you want to import the CA certificate, type `y` for yes.
- Restart Integration Server so it will use the new CA certificate.

Certificates Used for Secure Connections

My webMethods Server includes two default keystores you can use to set up and test your HTTPS listener:

- A *keystore*, which contains a key pair used to set up encrypted connections between client and server. The default keystore file is `localhost.p12`, which contains a demonstration certificate and a private key for the Jetty SSL server used by My webMethods Server.
- A *truststore*, which contains trusted digital certificates from a certification authority (CA). The default truststore file `sagdemoca.jks` contains a Software AG CA certificate that allows one instance of My webMethods Server to trust SSL (Secure Sockets Layer) connections from another instance, or from other webMethods products.

My webMethods Server stores these keystores at this location for each server instance:

`Software AG_directory \MWS\server\serverName\config\security\`

For production environments, you can use certificates from a commercial authority such as Verisign or use an internal authority.

Replacing Keystores

The `wrapper.conf` file sets the values the `javax.net.ssl` system properties use for communication using SSL. You can edit the `custom_wrapper.conf` file to replace the keystore or truststore for an instance of My webMethods Server.

> To replace the keystore or truststore

- Open the `custom_wrapper.conf` file for the server instance in a text editor. You can find the file at this location:

`Software AG_directory \profiles\MWS_serverName\configuration\`

- In the `custom_wrapper.conf` file, add or change the values of the SSL properties as needed and save the file:

```
# SSL Properties
set.JAVA_KEYSTORE=Software AG_directory\MWS\server\default\config
  \security\localhost.p12
set.JAVA_KEYSTORETYPE=pkcs12
set.JAVA_KEYSTORE_PASSWORD=encrypted_password
set.JAVA_TRUSTSTORE=Software AG_directory\MWS\server\default\config
  \security\sagdemoca.jks
set.JAVA_TRUSTSTORETYPE=jks
```

```
set.JAVA_TRUSTSTORE_PASSWORD=encrypted_password
```

- Restart My webMethods Server.

To provide an encrypted password for the keystore, see [“Generating an Encrypted Password” on page 35](#).

Generating an Encrypted Password

When replacing keystores, you must provide an encrypted password. Use the following procedure to generate the password, and then copy and paste it into the appropriate location in the `custom_wrapper.conf` file.

This procedure requires the use of the JDK, installed in the *Software AG directory*\jvm directory, or a later JDK version.

➤ To generate an encrypted password

- Open a command prompt window in the *Software AG directory* \jvm\jvm\bin\java directory.
- Run the following command:

- On Windows:

```
java -cp "Software AG directory\common\lib\wm-caf-common.jar;  
Software AG directory\common\lib\wm-scg-security.jar;  
Software AG directory\common\lib\wm-scg-core.jar;  
Software AG directory\common\lib\ext\slf4j-api.jar;  
Software AG directory\common\lib\ext\log4j*"   
com.webmethods.caf.common.CipherUtil password_to_encrypt
```

- On UNIX:

```
java -cp "Software AG directory/common/lib/wm-caf-common.jar:  
Software AG directory/common/lib/wm-scg-security.jar:  
Software AG directory/common/lib/wm-scg-core.jar:  
Software AG directory/common/lib/ext/slf4j-api.jar:  
Software AG directory/common/lib/ext/log4j/*"   
com.webmethods.caf.common.CipherUtil password_to_encrypt
```

The command returns an encrypted version of the specified password.

- Copy the encrypted password and paste it into the appropriate location in the `custom_wrapper.conf` file.

Communicating with webMethods Applications Using HTTPS

By default, My webMethods Server communicates with other webMethods applications using the HTTP protocol. You can change to the HTTPS protocol by doing the following:

➤ To cause My webMethods Server to communicate only through the HTTPS protocol

1. In My webMethods Server do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Advanced or Cluster Configuration.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Advanced or Cluster Configuration.**
2. In the **MWS Front End URL** field, modify the URL as follows:

- a. Change http to https.
- b. Change the port number.

For example, change this URL:

```
http://my_host:8585
```

To this:

```
https://my_host:7238
```

3. In the **HTTP Port** field, change the value to zero.

Note:

If running a cluster of My webMethods Servers, you must change the HTTP port value to zero for all nodes in the cluster.

4. In the **HTTPS Port** field, change the value to the port number and click **Submit**.
5. Restart My webMethods Server.
6. Notify administrators of all webMethods applications that communicate with My webMethods Server of the new HTTPS port number.

Managing Authentication Certificates as My webMethods Administrator

As My webMethods Administrator you can manage authentication certificates for users who connect to Integration Server or other webMethods applications. Authentication certificates do not govern a connection between a user and My webMethods Server. To be assigned a certificate, the user must be a member of the system directory service or an external directory service connected to My webMethods Server. For more information about working with directory services, see [“Managing Directory Services” on page 110](#).

The assignment of users to authenticates follows these rules:

- A user can be assigned to multiple certificates.
- An instance of a certificate can have only one user assigned to it, but you can add multiple instances of a certificate, each with a different certificate type, and assign a different user to each instance.

You can perform the following actions with certificates:

- Add a certificate.
- Search for a certificate.
- View details of a certificate.
- Assign a user to a certificate.
- Change users for a certificate.
- Delete a certificate.

Adding an Authentication Certificate

To add a certificate, do the following:

➤ To add an authentication certificate

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Certificates.**
2. Click **Add New Certificate.**
3. Click **Browse**, navigate to the location of the certificate file you want to add, and click **Open.**
4. From the **Certificate Type** list, choose the type authentication certificate to be used by a client connecting to Integration Server or other webMethods application:

The following table lists the types of certificates that My webMethods Server supports, and their purpose:

Certificate Type	Purpose
SSL (default)	Authenticates the message sender. The credentials are supplied in the protocol header.
Verify	Verifies the digital signature on incoming messages to Integration Server.
Encrypt	Encrypts outgoing messages from Integration Server.

Certificate Type	Purpose
Verify and Encrypt	Both verifies the digital signature on incoming messages and encrypts outgoing messages. Used if a user has the same certificate for sending and receiving messages.
Message Authentication	Authenticates the message sender. The credentials are supplied in the message header.

5. Click **Upload**.

The certificate appears on the Certificates panel.

Searching for Authentication Certificates

You can search for authentication certificates based on a number of criteria.

➤ To perform a search for authentication certificates

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Certificates > Search > Advanced**.
2. Specify the criteria to apply when searching for certificates:

The following table lists the available search criteria and how to configure the advanced search:

Field	Description
CERTIFICATE INFO	
Type	Choose the certificate type assigned to the certificate. For more information about valid types, see “Adding an Authentication Certificate” on page 37 . The default is Any .
Issuer Name	Type the common name of the certificate issuer. This field is not used if you leave it blank.
Serial Number	Type the serial number assigned to the certificate. This field is not used if you leave it blank.
Subject Name	Type the common name of the subject. This field is not used if you leave it blank.
VALID NOT BEFORE DATE	
Range	Choose a range of dates from the selection provided. The default is All .

Field	Description
Start Date	Type a start date using the format M/D/YYYY; if you use an incorrect format, the border turns red. Or click  .
End Date	Type an end date using the format M/D/YYYY; if you use an incorrect format, the border turns red. Or click  .
VALID NOT AFTER DATE	
Range	Choose a range of dates from the selection provided. The default is All .
Start Date	Type a start date using the format M/D/YYYY; if you use an incorrect format, the border turns red. Or click  .
End Date	Type an end date using the format M/D/YYYY; if you use an incorrect format, the border turns red. Or click  .

3. After you have selected search criteria, click **Search**.

Viewing Details of an Authentication Certificate

You can view the details associated with an authentication certificate.

> To view the details of an authentication certificate

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Certificates**.
2. If the certificate is not visible in the **Certificates** panel, use the **Search** panel to locate it. For more information, see [“Viewing Details of an Authentication Certificate” on page 39](#).
3. Locate the certificate and click .

The following table lists the details, available for each authentication certificate:

Certificate Detail	Description
Type	The certificate type assigned when the certificate was added.
Subject CN	The common name of the host being authenticated.
Issuer CN	The common name of the issuer.
Serial Number	The serial number assigned to the certificate.
Valid Not Before	The date before which the certificate is not valid.

Certificate Detail	Description
Valid Not After	The date after which the certificate is not valid.
User	The user's name.

4. To return to the list of certificates, click **Close**.

Assigning a User to an Authentication Certificate

You can assign only one user to an instance of an authentication certificate. The procedure for doing so is described here. To assign the same certificate to multiple users, add a separate instance of the certificate for each user. For more information on how to add a certificate, see [“Adding an Authentication Certificate” on page 37](#).

➤ To assign a user to an authentication certificate

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Certificates**.
2. If the certificate is not visible in the **Certificates** panel, use the **Search** panel to locate it. For more information, see [“Searching for Authentication Certificates” on page 38](#).
3. Locate the certificate and click .
4. On the Edit Certificate area, click **Set**.
5. In the Keywords field, type a user ID, click **Search**, move the user to the **Selected** box, and click **Apply**.
6. Click **Close**.

Changing Users for an Authentication Certificate

You can exchange one user for another in an existing authentication certificate.

➤ To change users for an authentication certificate

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Certificates**.
2. If the certificate is not visible in the **Certificates** panel, use the **Search** panel to locate it. For more information, see [“Searching for Authentication Certificates” on page 38](#).

3. Locate the certificate and click .
4. On the Edit Certificate area, click **Set**.
5. In the Keywords field, type the user ID for the new user, click **Search**, move the user to the **Selected** box, and click **Apply**.

This action replaces the former user with the new user.

6. Click **Close**.

Removing a User from an Authentication Certificate

You can remove a user who is assigned to an existing authentication certificate.

➤ To remove a user from an authentication certificate

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Certificates**.
2. If the certificate is not visible in the **Certificates** panel, use the **Search** panel to locate it. For more information, see [“Searching for Authentication Certificates” on page 38](#).
3. Locate the certificate and click .
4. On the Edit Certificate area, click **Remove**.
5. Click **Close**.

Deleting an Authentication Certificate

You can view the details associated with an authentication certificate.

➤ To view the details of an authentication certificate

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Certificates**.
2. If the certificate is not visible in the **Certificates** panel, use the **Search** panel to locate it. For more information, see [“Searching for Authentication Certificates” on page 38](#).
3. In the search results, select the check boxes beside the certificates you want to delete, and click **Delete**.

4 Using My webMethods Server with Web Servers

- Integration with Web Servers 44
- My webMethods Server and Apache 44
- My webMethods Server and IIS 44

Integration with Web Servers

My webMethods Server can integrate with the leading web servers, such as Microsoft Internet Information Server or Apache HTTP Server. The primary mechanism for integrating My webMethods Server with a third party web server in a distributed deployment scenario requires the use of a small plug-in that is installed and configured on the web server. This plug-in forwards HTTP requests from the web server to My webMethods Server.

My webMethods Server provides an integrated servlet engine with Jetty, which is a built-in web server that supports both HTTP and HTTPS. As such, having a separate web server tier is not a hard requirement.

There are several reasons for configuring My webMethods Server with an external web server (or cluster of web servers). The most notable reason is to adhere to corporate IT policies and procedures. My webMethods Server supports a flexible deployment model that allows an external web server (or cluster of web servers) to handle all HTTP requests that can be separately load balanced.

Integrating an external web server to handle HTTP requests requires configuring a web server plug-in on the external web server machine(s). The web server plug-in leverages code from the Jakarta web server project which is used extensively across many production-quality web server products.

My webMethods Server and Apache

You can use My webMethods Server on UNIX platforms with the Apache HTTP Server, from the Apache Software Foundation. If you want to use the Apache HTTP Server, you may find these external references useful:

- For information on using the Apache `mod_proxy_http` module: http://httpd.apache.org/docs/2.2/mod/mod_proxy_http.html
- For information on using the Apache `mod_proxy` module: http://httpd.apache.org/docs/2.2/mod/mod_proxy.html
- For information on using the Apache `mod_proxy_balancer` module: http://httpd.apache.org/docs/2.2/mod/mod_proxy_balancer.html

My webMethods Server and IIS

Software AG does not include Internet Information Services (IIS) components in an installation of My webMethods Server, nor does My webMethods Server explicitly support the use of IIS as a web server. However, if you want to use IIS, you may find these external references useful:

- For general information and discussions on using IIS as a reverse proxy: <http://www.iis.net/>
- To use IIS with Apache Tomcat: http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html

5 Using My webMethods Server with Docker

■ About My webMethods Server and Docker Containers	46
■ Running My webMethods Server in Docker Containers	46
■ Considerations when Working with My webMethods Server Containers	46
■ Creating and Customizing Docker Images for My webMethods Server Installations	47
■ Running and Customizing My webMethods Server Containers	50
■ About the Optimized My webMethods Server Container Startup	61
■ Using My webMethods Server Containers with webMethods Microservices Runtime	62
■ Monitoring My webMethods Server Containers	63
■ Clustering My webMethods Server Containers	68

About My webMethods Server and Docker Containers

Docker is a set of tools and services for building, deploying and managing applications. Docker provides virtualization through application packages, called containers. A container packages the application together with its entire runtime environment, including libraries, configuration files, and other dependencies. Containers run in the Docker environment as instances created from predefined images, and My webMethods Server provides a set of tools for creating and customizing My webMethods Server images for use with Docker, together with different options to modify the configuration of individual containers.

Running My webMethods Server in Docker Containers

The high-level steps for containerizing your My webMethods Server installation are as follows:

- Install My webMethods Server and the required layered products on Linux as described in [Installing SAG products](#).
- Create a My webMethods Server instance
- Run the My webMethods Server tooling to create a Dockerfile for the installation
- Build a Docker image for the installation using the generated Dockerfile
- Run containers from the image

When you use the Docker command line to create a new container from a My webMethods Server image, My webMethods Server uses a dedicated startup script to start a server instance inside the container. You can use environment variables to pass specific configurations to the startup script to override the predefined settings from the Docker image, for example to start My webMethods Server with a different database than the database, used by the instance from the image, or to redirect the server logs to a different storage location. For more information about the environment variables for My webMethods Server, see [“About the My webMethods Server Environment Variables” on page 50](#)

When starting up in a container, My webMethods Server performs a series of additional validation checks such as database connectivity, database integrity, and volume consistency checks. If any of the validation checks fails, the container exits. Unless you redirect the server logs to another location, logs are available on the console or in the default volumes that Docker creates for My webMethods Server. For more information about the default volumes, see [“Managing Assets and Data in My webMethods Server Containers” on page 53](#)

Considerations when Working with My webMethods Server Containers

Support for My webMethods Server with Docker 19.03.0 and later is available on Linux systems. For supported Linux systems, see the Docker documentation.

Software AG recommends that you create images and containers from fully configured My webMethods Server installations and install all additional applications and layered products before building a Docker image. For more information about Docker support for layered products, see their respective documentation.

You can only containerize single-instance installations. Container clustering is handled using Docker-native approaches and additional container orchestration technologies such as Kubernetes.

The My webMethods Server documentation assumes familiarity with Docker and the Docker terminology. For information about native Docker functionality, see the Docker documentation.

The My webMethods Server documentation assumes that the on-premise installation from which you create a Docker image uses the default paths and locations.

Creating and Customizing Docker Images for My webMethods Server Installations

My webMethods Server provides additional tooling to facilitate the generation of Dockerfiles and Docker images from your installation. The `mws-docker.sh` script is available in the `Software AG_directory /MWS/tools/docker/bin` directory of your My webMethods Server installation and includes a set of commands for creating Dockerfiles and images for My webMethods Server, and for working with image repositories.

Prerequisites for Building a Docker Image for a My webMethods Server Instance

Before you can create a Dockerfile and build a Docker image for a My webMethods Server instance, you must do the following:

- Install Docker and start the Docker daemon.
- On the same machine, install My webMethods Server and all required layered products, and create a new server instance. Optionally, initialize the instance.
- Install all required product fixes and custom applications.

You can create the server instance using Software AG Installer, or the My webMethods Server command-line utility.

Creating a Dockerfile for a My webMethods Server Installation

The `mws-docker.sh` script provides additional command-line tooling for generating a Dockerfile for a single-instance installation of My webMethods Server.

➤ **To generate a Dockerfile for My webMethods Server**

1. Go to the *Software AG_directory* `/MWS/tools/docker/bin` directory of your My webMethods Server installation.
2. Run `mws-docker.sh createDockerfile`, and customize the generated Dockerfile by specifying one or more of the following parameters:

Parameter	Description
<code>-Dfile.path</code>	Optional. The file path for the generated Dockerfile. Specify a full path, or a file name only. The default values are: <i>Software AG_directory</i> <code>/MWS/tools/docker/scripts/file.path</code> if you specify only a file name, and <i>Software AG_directory</i> <code>/MWS/tools/docker/scripts/Dockerfile_MWS</code> if not specified.
<code>-Dimage.name</code>	Optional. The name of the base image on top of which to build the My webMethods Server image. The default is <code>centos:7</code> .
<code>-Dcreate.user.group</code>	Optional. Boolean. Whether to create a user and group for the image. The default is <code>true</code> .
<code>-Dbase.user.group</code>	Specify as <code>userid:groupid</code> . Optional. Use when <code>-Dcreate.user.group</code> is set to <code>false</code> . Specify the user and group to supply to the <code>chown</code> statements in the generated Dockerfile and use when copying files/directories from the local installation to the image. You must make sure that the specified user and group exist, i.e. your base image or additional scripts must create the user/group.
<code>-Dinstance.name</code>	Optional. The name of the My webMethods Server instance for which to create an image. The default is <code>default</code> .
<code>-Dinclude.jdk</code>	Optional. Boolean. Whether to include the JDK from the host installation in the image. The default is <code>true</code> .
<code>-Dimage.java.home</code>	Optional. Full path to the java home directory. Applies only when <code>-Dinclude.jdk</code> is set to <code>false</code> .
<code>-Dport.list</code>	Optional. A comma-separated list of ports to expose in the Docker container. The default is <code>8585</code> .
<code>-Dcreate.volumes</code>	Optional. Boolean. Whether to set up volumes. The default is <code>true</code> .
<code>-Dcreate.dockerignore</code>	Optional. Boolean. Whether to generate a <code>.dockerignore</code> file. The default is <code>true</code> .

Building a Docker Image from a My webMethods Server Installation

The `mws-docker.sh` script provides additional command-line tooling for using the `docker build` command to create an image for a single-instance installation of My webMethods Server from a provided Dockerfile.

➤ To build a Docker image for My webMethods Server

1. Go to the *Software AG_directory* `/MWS/tools/docker/bin` directory of your My webMethods Server installation.
2. Run `mws-docker.sh build`, and customize the image generation by specifying one or more of the following parameters:

Parameter	Description
<code>-Duse.dockerignore</code>	Optional. Boolean. Specifies whether to use a <code>.dockerignore</code> file. The default is <code>true</code> .
<code>-Dfile.path</code>	Optional. The file path of the Dockerfile to use for the build. Specify a full path, or a file name only. The default values are: <i>Software AG_directory</i> <code>/MWS/tools/docker/scripts/file.path</code> if you specify only a file name, and <i>Software AG_directory</i> <code>/MWS/tools/docker/scripts/Dockerfile_MWS</code> if not specified.
<code>-Dimage.name</code>	Optional. A name for the generated My webMethods Server image. The default is <code>mws-image</code> .
<code>-Dserver</code>	Optional. URL to the Docker registry. If not specified, the script uses the local Docker registry.
<code>-Duser</code>	Optional. The user name of the user to authenticate to the Docker registry. Required when <code>-Dserver</code> is specified and the registry requires authentication.
<code>-Dpassword</code>	Optional. The password of the user to log on to the Docker registry. Required when <code>-Dserver</code> is specified and the registry requires authentication. Specify a plain text password or the full path to a text file that contains the password, for example <code>-Dpassword=file://full_file_path</code> .

Other Docker Image Commands

The `mws-docker.sh` script provides the following additional commands for working with Docker images for My webMethods Server.

savelmage

Saves an image from the local Docker registry to a zip file. This command accepts the following parameters:

- `-Dimage.name`: Required. The name of the image to save.
- `-Dfile.path`: Optional. The file path for the image. The default values are: *Software AG_directory* `/MWS/tools/docker/images/file.path` if you specify only a file name, and *Software AG_directory* `/MWS/tools/docker/images/image.name.zip` if not specified.

loadImage

Loads an image from an archive into the local Docker registry. This command accepts the following parameters:

- `-Dfile.path`: Required. The file path for the image to load. Specify an absolute path or a file name only. If you specify only a file name, the image is loaded from `Software AG_directory /MWS/tools/docker/images/file.path`.

pushImage

Pushes a My webMethods Server image to a remote Docker registry. This command accepts the following parameters:

- `-Dimage.name`: Required. The name of the My webMethods Server image to push.
- `-Dimage.tag`: Optional. tag name to use for the image when pushing it to the Docker registry. The default is `image.name`.
- `-Dserver`: Required. URL to the Docker registry.
- `-Drepository.name`: Required. The name of the repository to which to push the image.
- `-Duser`: Required. The user name of the user to authenticate to the Docker registry.
- `-Dpassword`: Required. The password of the user to log on to the Docker registry. Specify a plain text password or the full path to a text file that contains the password, for example `-Dpassword=file://full_file_path`.

Running and Customizing My webMethods Server Containers

About the My webMethods Server Environment Variables

When starting a new container from a My webMethods Server image you can override the default configuration of the image using environment variables. My webMethods Server container environment variables match the corresponding parameters of the `mws new` command, or provide container-specific configuration options.

You supply My webMethods Server environment variables using the `-e/--env` parameters of the `docker run` command, or in a file. For more information, see the Docker documentation.

Environment Variables that My webMethods Server Containers Support

My webMethods Server exposes the following environment variables for overriding the default configuration of an existing My webMethods Server image:

Variable	Description
DB_TYPE	The type of database used by the server instance. Valid values are: <ul style="list-style-type: none"> ■ ms - Microsoft SQL Server ■ oracle - Oracle ■ db2 - DB2 ■ mysql_ee - MySQL Enterprise Edition ■ mysql_ce - MySQL Community Edition ■ postgresql - PostgreSQL
DB_URL	Database connection URL, based on the type of database and the driver. Enclose the database URL in double quotes when supplying the variable via the Docker command line.
DB_USERNAME	The user name to use when connecting to the My webMethods Server database.
DB_PASSWORD	The password of the My webMethods Server database user.
NODE_NAME	A custom name for the cluster node that hosts the server instance.
APPS_DIR	A directory on the container file system which contains the custom assets and applications to copy to the <code>MWS/dep/loy</code> directory of the My webMethods Server instance, and installed on container startup. If not specified, My webMethods Server uses the default <code>SAGHOME/MWS/volumes/apps</code> directory in the container. Assets and applications are sourced through a bind-mounted host directory, or an external volume, mounted to the <code>apps</code> directory location in the container. Valid asset formats are <code>war</code> , <code>cdp</code> , <code>pdp</code> , <code>jar</code> . For more information about deploying applications to a My webMethods Server Docker container, see “Custom Applications in My webMethods Server Containers” on page 55
CONFIGS_DIR	A directory on the container file system which contains miscellaneous configuration files that My webMethods Server loads on container startup. If not specified, the container startup script for My webMethods Server uses the default <code>SAGHOME/MWS/volumes/configs</code> and checks for a <code>configs</code> directory on the volume, mounted to the <code>SAGHOME/MWS/volumes/configs</code> directory on the container file system if such volume is available. <p>The <code>configs</code> directory on the mounted volume can contain one or more of the following subdirectories:</p> <ul style="list-style-type: none"> ■ <code>assets_cfg</code> - for supplying <code>xmlImport</code> files and My webMethods Server skins. ■ <code>cluster_cfg</code> - stores custom configuration files for the My webMethods Server cluster, for example the <code>cluster.xml</code> file.

Variable	Description
	<ul style="list-style-type: none"> ■ <code>instance_cfg</code> - stores custom configuration files for the My webMethods Server instance, for example the <code>mws.db.xml</code> and <code>server.properties</code> files. ■ <code>jvm_cfg</code> - for supplying custom JVM configuration files and certificates. ■ <code>profile_cfg</code> - stores custom configuration files for the My webMethods Server OSGi profile, for example the <code>custom_wrapper.conf</code> file. <p>For more information about modifying the configuration of a My webMethods Server Docker container through an external volume, see “Modifying the Configuration of a Container” on page 58</p>
DATA_DIR	<p>Directory on the container file system to which My webMethods Server stores runtime data, such as search indexes and information about the deployed applications, and persists events from the Task Engine event queue. If not specified, My webMethods Server uses the default <code>SAGHOME/MWS/volumes/data</code> directory on the container file system.</p>
LIBS_DIR	<p>Directory on the container file system that holds third-party libraries or other custom jar files to be copied to the <code>MWS/lib</code> directory of the My webMethods Server instance and loaded by My webMethods Server on container startup. If not specified, the container startup script for My webMethods Server uses the default directory <code>SAGHOME/MWS/volumes/libs</code> on the container file system. Libraries and jars are sourced through a bind-mounted host directory or an external volume. For more information about using third-party libraries and custom jars in a My webMethods Server Docker container, see “Using External Libraries in My webMethods Server Containers” on page 57.</p>
LOGS_DIR	<p>Directory on the container file system to which My webMethods Server persists all log files, generated by the My webMethods Server instance. By default, the directory contains the following subdirectories:</p> <ul style="list-style-type: none"> ■ <code>instance_logs</code> - corresponds to the default <code>SAGDir/MWS/server/instanceName/logs</code> directory of an on-premise My webMethods Server installation. ■ <code>profile_logs</code> - corresponds to the <code>SAGDir/profiles/MWS_default/logs</code> directory of an on-premise My webMethods Server installation. ■ <code>cli_logs</code> - for all logs, stored in the <code>SAGDir/MWS/bin</code> directory of an on-premise installation. Logs for OSGi profile-related operations that My webMethods Server executes during operation, for example, when adding external libraries, are stored in <code>cli_logs/archive</code>. <p>If not specified, My webMethods Server uses the default directory <code>SAGHOME/MWS/volumes/logs</code> on the container file system.</p>

Examples

- The following example starts a new My webMethods Server container based on the `myMWSImage` image and specifies a custom node name for the cluster node that corresponds to the My webMethods Server instance:

```
docker run --env NODE_NAME=myClusterNode myMWSImage
```

- The following example starts a new My webMethods Server container based on the `myMWSImage` image and overrides the default database connection parameters of the image:

```
docker run --env DB_TYPE=mysql --env
DB_URL="jdbc:mysql://host:port;databaseName=wmdb;relaxAutoCommit=true"
--env DB_USERNAME=wmuser DB_PASSWORD=wmpass myMWSImage
```

At startup the My webMethods Server instance connects to a MySQL Enterprise Edition database with the name `wmdb`, on the specified `host` and `port`, using the credentials `wmuser/wmpass`, regardless of the default database configuration of the `myMWSImage` image.

- The following example starts a new My webMethods Server container based on the `mws` image, overrides the default database connection parameters for the image, and maps the default My webMethods Server connection port to an exposed container port:

```
docker run -e DB_URL="jdbc:wm:sqlserver://mwssql2016:1433;databaseName=mwsDB"
-e DB_USERNAME=wmuser -e DB_PASSWORD=wmpass -e DB_TYPE=ms -p 8585:8585 mws
```

At startup the My webMethods Server instance connects to a MS SQL Server database with the name `mwsDB`, on the host `mwssql2016` and port `1433`, using the credentials `wmuser/wmpass`, regardless of the default database configuration of the `mws` image. The My webMethods Server connection port `8585` is mapped to a port with the same number on the container host machine.

- The following example redirects the log file storage location for a My webMethods Server container based on the `myMWSImage` to the `myLogs` directory on the container file system, and persists all logs to the mounted directory `MWSLogs` on the host:

```
docker run -v /home/myDirs/MWSLogs:/volumes/myVolume/myLogs --env
LOGS_DIR=/volumes/myVolume/myLogs myMWSImage
```

Managing Assets and Data in My webMethods Server Containers

By default My webMethods Server containers persist data to the `SAGHOME/MWS/volumes` directory of the container, declared in the Dockerfile for My webMethods Server images. Unless you map a named volume, the Docker daemon mounts the `MWS/volumes` directory to an anonymous volume, that is, a volume with a random name, in the default location on the host file system `var/lib/docker/volumes`. You can use named volumes to facilitate archiving, navigation, and data sharing among different containers.

Although a My webMethods Server container can connect to a non-empty volume, unexpected results might occur if you connect a My webMethods Server container to a volume that already contains the data of another container. My webMethods Server validates the volume content at container start up, and stops the container if the validation fails. The following table lists the

volumes that the Dockerfile for My webMethods Server declares, and what restrictions apply to reusing volumes between containers:

Volume	Environment Variable	Description
MWS/volumes/apps	\$APPS_DIR	Stores custom applications to deploy to the My webMethods Server instance at startup. Read mode. Can be reused between different containers.
MWS/volumes/configs	\$CONFIGS_DIR	Stores configuration files to supply to the My webMethods Server instance at startup. Read mode. Can be reused between different containers.
MWS/volumes/data	\$DATA_DIR	Stores container runtime data, such as the My webMethods Server search index and events from the Task Engine lightweight event queue. Read and write mode. You cannot reuse this volume between different containers.
MWS/volumes/libs	\$LIBS_DIR	Stores external libraries and custom jars to inject to the MWS/lib directory at startup. Read mode. Can be reused between different containers.
MWS/volumes/logs	\$LOGS_DIR	Stores container runtime logs. Read and write mode. You cannot reuse this volume between different containers.
MWS/volumes/patches	\$PATCHES_DIR	Stores test patches to install on the My webMethods Server instance at container startup. Read mode. Can be reused between different containers.

For more information about changing the default asset and data locations for My webMethods Server container, see [“Environment Variables that My webMethods Server Containers Support” on page 50](#)

About Volumes and Volume Validation

In addition to the database connectivity and integrity checks, My webMethods Server also performs volume validation when you start a new container from a My webMethods Server Docker image. Volume validation checks ensure the following:

- The specified paths to bind-mounted host directories exist
- The specified named volumes and host directories have appropriate read/write permissions
- Only volumes and directories that containers can share are prepopulated with data.

If any of the volume validation checks fails, the container exits. Logs are available on the console and in the default (unnamed) Docker volume. For more information about volume and data sharing between containers, see [“Managing Assets and Data in My webMethods Server Containers” on page 53](#).

Installing My webMethods Server Components and Custom Applications

Similar to an on-premise My webMethods Server installation, you can use the Install Administration portlet to install My webMethods Server components to an instance running in a Docker container. My webMethods Server marks all server components that you install on top of the Docker image from which a container is created with the following icon .

You can also remove server components installed on top of the image using the Install Administration portlet. However, you cannot browse for custom applications and deploy them. To deploy custom applications to a My webMethods Server container, see [“Custom Applications in My webMethods Server Containers” on page 55](#).

For general information about the Install Administration portlet, see [“Installing Portlets or Other Deployable Server Components” on page 330](#).

Custom Applications in My webMethods Server Containers

Despite the general recommendation to include all required custom applications before generating a Docker image for a My webMethods Server installations, in certain occasions and especially for non-production environments you might require to install a custom application in a container on top of the image from which you start the container. My webMethods Server containers include a predefined directory `/MWS/volumes/apps` on the container file system, in which you can place assets to copy to the `/deploy` directory of the My webMethods Server instance. You can install custom applications in a container in one the following ways:

- Using the `docker cp` command - to copy the application package from the host machine to the `/MWS/volumes/apps` directory of a running container. This approach hot deploys the application and does not require restarting the container. However, you cannot use this approach if your application includes dependencies to external libraries. You cannot remove or uninstall an application that you hot-deploy using the `docker cp` command.
- Using a bind-mounted directory from the host files system - use the `-v` option of the `docker run` command to bind mount a directory from the host that contains one or more applications to deploy when starting up a new container. At container startup My webMethods Server copies all files from the host directory to the `/MWS/volumes/apps` of the container, and subsequently - to the `/deploy` directory for the instance. To remove an application, delete it from the mounted directory and restart the container.
- Using a named volume - use the `-v` option of the `docker run` command to mount a named volume that contains one or more applications to deploy when starting up a new container. At container startup My webMethods Server copies all files from the volume to the `/MWS/volumes/apps` of the container, and subsequently - to the `/deploy` directory for the instance. To remove an application, delete it from the volume and restart the container.

My webMethods Server persists a list of all deployed applications in a `.properties` file on the `MWS/volumes/data` directory in the container and its mounted volume, which is anonymous by default.

You can override the container locations of both the `/apps` and the `/data` directories, using the environment variables for My webMethods Server. For more information, see [“Environment Variables that My webMethods Server Containers Support” on page 50](#).

Examples

- The following example creates the named volume `tasks` and uses it to deploy task applications to a container, created from the `webM107` image:

```
docker volume create tasks
```

```
docker run -e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms -v tasks:/opt/softwareag/MWS/volumes/apps -p 8585:8585 webM107
```

Note:

The custom task applications are copied to the volume from the host before issuing the `docker run` command.

- The following example uses bind-mounts the host directory `/home/myvm/taskapps` to deploy task applications to a container, created from the `webM107` image:

```
docker run -e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms -v
/home/myvm/taskapps:/opt/softwareag/MWS/volumes/apps -p 8585:8585
webM107
```

Note:

The custom task applications must be present in the bind-mounted directory.

- The following example uses bind-mounts the host directory `/home/myvm/taskapps` to deploy task applications to a container, created from the `webM107` image and copies all applications to the `/tasks` directory in the container, instead of in the default `/apps` directory:

```
docker run -e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms -v /home/myvm/taskapps:/opt/softwareag/MWS/tasks -e
APPS_DIR=/opt/softwareag/MWS/tasks -p 8585:8585
webM107
```

Note:

The custom task applications must be present in the bind-mounted directory. The `MWS/tasks` directory, referenced by the `APPS_DIR` environment variable is created at container startup.

Deploying Assets Using `xmlImport` Files

My webMethods Server containers can use a dedicated location `/MWS/volumes/configs/assets_cfg` on the container file system to which you can mount an external volume and supply `xmlImport` files that create and modify My webMethods Server assets such as user interface elements, folders, users, and so on during container startup.

To use the `/MWS/volumes/configs/assets_cfg` to deploy `xmlImport` files, you must mount an external volume to the `MWS/volumes/configs` directory, and that volume must contain a directory with the name `assets_cfg`. If you do not mount an external volume, or the volume does not include an `assets_cfg` directory, the Docker daemon does not create an `/MWS/volumes/configs/assets_cfg` subdirectory on the container file system.

Redirecting Container Logs to a Non-Perishable Storage

When you create a new container from a My webMethods Server image, the startup script for My webMethods Server maps the `wrapper.log` to the standard output stream (STDOUT) for the container. At runtime, the My webMethods Server instance stores all logs in the default container log directory `MWS/volumes/logs` directory on the container file system. This directory is mapped by default to an anonymous volume, created with a random name by the Docker daemon. When the container stops or gets restarted, the log files from previous runs are sourced back to the `MWS/volumes/logs` directory from the anonymous volume. You can map a directory from the host file system, or create a named volume persist the files to a non-perishable location and keep the log data for auditing purposes.

Additionally, you can change the log location on the container file system using the `LOGS_DIR` environment variable. For more information, see [“Environment Variables that My webMethods Server Containers Support” on page 50](#).

Using External Libraries in My webMethods Server Containers

You can add third-party libraries, custom jar files, and fragment bundles to My webMethods Server when starting up a new container. By default, My webMethods Server containers provide a directory `SAGHOME/MWS/volumes/libs` to which you can mount an external volume. When you create a new My webMethods Server container, the startup script for My webMethods Server copies all third-party libraries and custom jar files, available in the mounted volume to the `MWS/lib` directory to load when My webMethods Server initializes.

You can override the default location of the `SAGHOME/MWS/volumes/libs` directory within the container using the `LIBS_DIR` environment variable on container startup. If the directory referenced by the variable does not exist on the container file system, the Docker daemon creates the directory.

To customize the manifest of a third-party library, or add a jar file as a fragment to another bundle, provide a bind instruction as described in [“Adding Custom JAR Files” on page 345](#) and place the bind file in the volume together with the jar file.

Examples

The following example starts a new container from the `myMWSImage` and mounts an external volume to the default `SAGHOME/MWS/volumes/libs` directory on the container file system as a source for external libraries to be loaded by My webMethods Server when the instance initializes:

```
docker run -v var/lib/docker/volumes/myMWSVolume/test/extlibs:SAGHOME/MWS/volumes/libs
-p 8585:8585 myMWSImage
```

The following example starts a new container from the `myMWSImage`, mounts an external volume to the `opt/softwareag/MWS/myLibsDir` directory on the container file system and overrides the

default `SAGHOME/MWS/volumes/libs` directory with the `opt/softwareag/MWS/myLibsDir` directory as a source for external libraries to be loaded by My webMethods Server.

```
docker run -v
var/lib/docker/volumes/myMWSVolume/test/extlibs:opt/softwareag/MWS/myLibsDir --env
LIBS_DIR=opt/softwareag/MWS/myLibsDir -p 8585:8585 myMWSImage
```

In both examples, all libraries and custom jars from the mapped volume are copied to the `MWS/lib` directory and loaded by My webMethods Server on startup.

Modifying the Configuration of a Container

In addition to the `assets_cfg` subdirectory that you can use for deploying My webMethods Server assets, the `/MWS/volumes/configs` directory on the container file system can contain dedicated subdirectories for supplying miscellaneous configuration files for My webMethods Server to load at startup.

To use the `/MWS/volumes/configs/` directory to supply configuration files, you must mount an external volume to that directory, and the volume must contain one or more of the following subdirectories:

- `cluster_cfg` - for supplying custom configuration files such as `cluster.xml`, `systemProperties.properties`, and so on. The My webMethods Server that initializes the cluster database applies the settings from the files in the `cluster_cfg` directory on container bootstrap, across all nodes in the My webMethods Server cluster.
- `instance_cfg` - for supplying a custom `cache.xml`, `mws.db.xml`, `server.properties`, `websso.properties`, `log4j.init.properties`, and other instance configuration files.
- `jvm_cfg` - for supplying miscellaneous JVM/JRE additions, such as security certificates, to the container JRE.
- `profile_cfg` for supplying custom profile configurations, for example JVM properties in a `custom_wrapper.conf` file, or security settings in a `jaas.config` file.

The Docker daemon creates a corresponding `*_cfg` subdirectory on the container file system only for the `*_cfg` folders that are present in the mounted volume (if such volume is available).

Configurations that you supply using the environment variables for My webMethods Server containers always take precedence and replace their corresponding values in the configuration files, supplied through a volume.

Note:

Although you can modify the configuration files for My webMethods Server, such as the `cluster.xml`, or `custom_wrapper.conf` files, and supply their new versions from a volume when starting a new container from an existing image, Software AG recommends that you fully configure My webMethods Server before generating a Docker image from an on-premises installation.

Container Configuration Examples

Example: Configure the Database Connection Retries for a My webMethods Server Container

To configure the database connection retries for a My webMethods Server container, supply a new `mws.db.xml` file from an external volume.

1. Copy the `mws.db.xml` from an on-premises installation or a My webMethods Server container.
2. Modify the file to include the required settings for database connection retries as described in [“Modifying Database Connection Retries” on page 215](#). If required, modify connection parameters such as database connection URL and user password to supply correct configuration at container start up.
3. Place the `mws.db.xml` file in an `my_mws_config/instance_cfg` directory on the host file system and mount it to the `configs` directory in the container on start up, as follows:

```
docker run
-v /home/myvm/my_mws_config:/opt/softwareag/MWS/volumes/configs
-p 8586:8585 webm107
```

My webMethods Server will use all settings, defined in the `mws.db.xml` file from the mounted host directory, unless you override a setting using the environment variables for My webMethods Server containers. Environment variables that you supply using the Docker CLI overwrite their corresponding values in the `mws.db.xml` file from the mapped directory.

Example: Add Security Certificates to a My webMethods Server Container

You can add new security certificates to the container JRE when starting up a new My webMethods Server container. The high-level steps for adding security certificates are as follows:

1. Modify the `cacerts` file to add the new certificates using the `keytool` command on an on-premises installation that has the new certificates, or in a container that has access to the new certificate files, then extract the updated `cacerts` file.
2. On the host file system, create a dedicated directory that matches the following pattern: `my_mws_config/jvm_cfg/lib/security/`, and place the updated `cacerts` file in that directory.
3. Run the container with the `my_mws_config` directory, mapped to the `configs` directory on the container file system, as follows:

```
docker run
-e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms
-v /home/myvm/my_mws_config:/opt/softwareag/MWS/volumes/configs
-p 8585:8585 webM107
```

The sample command creates a My webMethods Server container from the `webM107` image. The container uses an MS SQL server database with the name `MWSDOCKER` and has additional security

certificates supplied through the `/home/myvm/my_mws_config/jvm_config/lib/security` directory on the host file system.

Example: Configure Single Sign-on for a My webMethods Server Container

The following example provides high-level steps for configuring single sign-on with a third-party identity provider for a My webMethods Server instance running in a Docker container, using the system-wide CA certificate store `cacerts` as the SSO keystore. The instructions assume that you are familiar with Docker networking and have configured your Docker environment. For more information about container networking, see the Docker documentation.

1. Start a container from the same image that you plan to use for running My webMethods Server containers with single sign-on.

Note:

This step is required in order to acquire and modify the configuration files for My webMethods Server. However, since starting a My webMethods Server container initializes the My webMethods Server database, you must either start the container with another database instance, or change the container entrypoint with the `docker run` command. For more information about container entrypoints, see the Docker documentation.

2. Navigate to the `/opt/softwareag/MWS/bin` directory of the running container and execute `sh mws.sh getConfig fileName` to retrieve the `cluster.xml`, and the `websso.properties` files.
3. Modify the `cluster.xml` file to expose an HTTPS port for My webMethods Server and place it in a `my_mws_config/cluster_cfg/` directory on the host file system.
4. Add the security certificate of the external IDP by modifying the `cacerts` file with the `keytool` command, then extract the updated `cacerts` file. If your container does not have access to the new certificate file(s), perform this step in advance on an on-premises installation that has the new certificates.
5. On the host file system, create a dedicated directory that matches the following pattern: `my_mws_config/jvm_cfg/lib/security/`, and place the updated `cacerts` file in that directory.
6. Modify the `websso.properties` file with the required properties and place it in a `my_mws_config/instance_cfg/` directory on the host file system. For more information, see [“Setting Properties in the websso.properties File” on page 282](#).

Tip:

To apply the configuration cluster-wide, place the `websso.properties` file in a `my_mws_config/cluster_cfg/` directory.

7. Run the container and map the parent directory `my_mws_config` that contains all directories with custom configuration files to the `configs` directory on the container file system, as follows:

```
docker run
-e
DB_URL="jdbc:wm:sqlserver://dbhost:1439;databaseName=MWSDOCKER;user=wmuser;password=manage"
-e DB_TYPE=ms
-v /home/myvm/my_mws_config:/opt/softwareag/MWS/volumes/configs
-p 8585:8585 webM107
```

On startup, My webMethods Server generates an `SPMetadata.xml` and an `IDPMetadata.xml` files in the `/opt/softwareag/MWS/server/instanceName/config/` directory in the container.

8. Complete the configuration by registering My webMethods Server with the external identity provider using the xml files, generated on the previous step. For more information, see [“Using Single Sign-On with SAML and a Third-Party Identity Provider”](#) on page 280.
9. Restart the My webMethods Server container.

About the Optimized My webMethods Server Container Startup

When initializing inside a container My webMethods Server uses a mechanism for automatic optimization of the container startup time. The optimization mechanism triggers when the average estimated average execution time of each SQL statement, required for initializing the instance exceeds 20 milliseconds due to network latency between My webMethods Server and the production database. The mechanism uses an in-memory instance of the Derby database, and executes all initial bootstrap activities against that Derby instance. Subsequently, all required data is transferred to the production database. This mechanism applies automatically for all containers that are estimated to execute their startup queries for a longer time than the specified threshold, regardless of the type of the production database.

You can control the SQL execution time threshold at which My webMethods Server triggers the startup optimization, or whether the optimization mechanism triggers at all using the following additional JVM properties:

- `mws.inmemory.init`: controls whether a My webMethods Server container uses the startup optimisation mechanism. By default, the property is not present in the `custom_wrapper.conf` file, and My webMethods Server executes a sample set of database queries, and initializes using the optimization mechanism only if the average execution time per query exceeds 20 milliseconds. Set to `true` to always use the optimization mechanism, or to `false` to initialize using the production database, regardless of the SQL execution time.
- `sql.delay.threshold`: the maximum average estimated amount of time for which an SQL query can execute during initialization for My webMethods Server to start up without using the optimization algorithm. Applies when the `mws.inmemory.init` property is not set to `true` or `false`. The default value is 20 milliseconds.

Supply the required properties as custom JVM parameters in a `custom_wrapper.conf` file from a mapped volume, For more information, see [“Modifying the Configuration of a Container”](#) on page 58.

Note:

Unless you switch off the startup optimization mechanism, all My webMethods Server containers require a local Derby database instance for optimized startup. The Derby scripts are available in the My webMethods Server installation, in `Software AG_directory/MWS/server/derby-scripts.zip`. When creating custom images from a My webMethods Server installation, make sure that the folder is not skipped, for example by a `.dockerignore` file that you are using.

Using My webMethods Server Containers with webMethods Microservices Runtime

webMethods Microservices Runtime is a lightweight container for running microservices. Microservices Runtime provides a subset of the features of Integration Server.

Certain limitations apply when you use My webMethods Server containers with Microservices Runtime as a web service provider. For example, since Microservices Runtime does not support SAML authentication, all web service connectors for custom applications and layered products, running on My webMethods Server must switch to basic authentication.

Basic authentication for CAF web service clients is enabled by default when My webMethods Server is running in a container. You can configure the user name and password with which web service clients authenticate against Microservices Runtime or Integration Server on the **CAF Application Runtime Configuration** page. For more information, see [“Configuring Web Service Authentication for CAF Applications” on page 62](#).

For more information about Microservices Runtime, see *Developing Microservices with webMethods Microservices Runtime*. For more information about running webMethods Microservices Runtime in a Docker container, see *webMethods Integration Server Administrator’s Guide*.

Configuring Web Service Authentication for CAF Applications

When My webMethods Server is running in a container, basic authentication is enabled and switched on by default for all web service clients in custom applications and layered products, overriding the settings of the individual application. You can disable this setting, for example, when using My webMethods Server containers with Integration Server containers, or modify the default user name and password for basic authentication.

➤ To modify the web service authentication settings for CAF applications

1. As sysadmin, go to **Folders > Administrative Folders > Administration Dashboard > Configuration > CAF Application Runtime Configuration**.
2. Click **Configure Global Defaults**, and then click **Environment Entries**.
3. Modify one or more of the following environment entries:

Environment Entry	Description
<code>wsclient.global.basicauth-enabled</code>	Whether to enable basic authentication globally for all web service clients. The default is <code>true - enabled</code> .
<code>wsclient.global.basicauth-username</code>	The user name to supply when authenticating to Microservices Runtime or Integration Server. The default is <code>Administrator</code> .

Environment Entry	Description
	<p>Note: This user is disabled by default.</p>
<code>wsclient.global.basicauth-password</code>	<p>The password to supply when authenticating to Microservices Runtime or Integration Server. The default is <code>manage</code>.</p> <p>Note: If using the default Administrator account, you must change this password with the password you provided when enabling the Administrator user.</p>

- Click **Apply**.

Monitoring My webMethods Server Containers

By default, the generated Dockerfile for My webMethods Server includes a `HEALTHCHECK` instruction that checks the status of each container by testing a number of health indicators such as database connectivity, disk space and memory status, directory service connectivity, and so on.

The container health check for My webMethods Server is configured to run 5 minutes after you start a container, to trigger each individual check 15 seconds after the previous check completes, and to fail if an individual check fails to complete within 30 seconds. You can modify these settings in the generated Dockerfile for your installation. For more information about modifying the `HEALTHCHECK` instruction, see the Docker documentation.

My webMethods Server containers include the following predefined health indicators:

- `MemoryHealthIndicator` - passes if the total amount of available memory in the system is larger than a predefined value. The predefined value is 20 (percent). Enabled by default.
- `JettyThreadHealthIndicator` - passes if the ratio of the number of the current Jetty threads to the number of the maximum available Jetty threads is larger than a predefined value. The predefined value is 20 (percent). Enabled by default.
- `DiskspaceHealthIndicator` - passes if the total amount of available disk space is larger than a predefined value. The predefined value is 20 (percent). Enabled by default.
- `DatabaseConnectivityHealthIndicator` - passes if the container can connect to the database and execute a simple query. Enabled by default.
- `AXSRFTGenerationHealthIndicator` - passes if My webMethods Server successfully generates an AXSRFT token. Enabled by default.
- `CustomDataSourceHealthIndicator` - passes if the container can connect to the external data sources, defined for My webMethods Server. Enabled by default.

- `DirServiceConnectivityHealthIndicator` - passes if the container can connect to the external data source(s), defined for My webMethods Server. Enabled by default.
- `ProcessEngineConnectivityHealthIndicator` - passes if the container can connect to the Integration Server that hosts Process Engine. Enabled by default for containers that include Task Engine.
- `RulesEngineConnectivityHealthIndicator` - passes if the container can connect to the Integration Server that hosts Business Rules. Enabled by default for containers that include Task Engine.
- `UmConnectionHealthIndicator` - passes if the container can connect to the Universal Messaging. Enabled by default when `useDbJms=false`.

My webMethods Server containers provide a dedicated endpoint at `http://container_host:port/health`, which exposes the current values, measured by all configured health indicators. Accessing the `health` endpoint requires basic authentication with an account that is a member of the Admin role.

You can configure the default indicator thresholds, and switch individual checks on or off using a `healthCheck.json` configuration file. For more information about the healthcheck configuration file, see [“Container Health Check Samples” on page 65](#).

In addition to the default health indicators for containers, you can implement a custom indicator and register it with My webMethods Server, as described in [“Implementing Custom Health Indicators” on page 64](#).

Implementing Custom Health Indicators

To add a custom health indicator in a My webMethods Server container, you must register your implementation as an OSGI service. On container start up, My webMethods Server loads custom health indicators from the `MWS/volumes/libs` directory of the container (if included in the image), or from a mapped directory on the host file system, and lists them at the dedicated endpoint `http://container_host:port/health` together with the built-in health indicators.

The high-level steps to create and register a custom health indicator for My webMethods Server containers are as follows:

- In an application project, create a class that implements `IHealthIndicator`.
- Add the custom code that performs the required health checks to the class.
- In the same application project, add a custom class that implements the `IHealthIndicatorConfig` class, with the custom code that configures your indicator properties.
- Package the application as a jar file.
- Add the new custom indicator in a `healthCheck.json` file, as described in [“Container Health Check Samples” on page 65](#).
- Place the `healthCheck.json` file in an `instance_cfg` directory, and the compiled application in another dedicated directory, both in a volume or on the host file system.

- When starting the My webMethods Server container, mount the `instance_cfg` to the `MWS/volumes/config` directory on the container file system, and the directory that contains the CAF application to `MWS/volumes/libs`.

Container Health Check Samples

Example: healthCheck.json

Create a `healthCheck.json` file using the following example and include your custom health indicator together with the default indicators. The configuration that you specify with a `healthCheck.json` file overrides all health check indicator settings. Changes that you make to the `healthCheck.json` file apply when starting or restarting the container.

```
{
  "enabled": "true",
  "indicators": {
    "MemoryHealthIndicator": {
      "enabled": "true",
      "properties": {
        "memoryAvailable": 20
      }
    },
    "JettyThreadHealthIndicator": {
      "enabled": "true",
      "properties": {
        "currentToMaxThreadRatioThreshold": 20
      }
    },
    "DiskSpaceHealthIndicator": {
      "enabled": "true",
      "properties": {
        "diskSpaceAvailable": 20
      }
    },
    "DatabaseConnectivityHealthIndicator": {
      "enabled": "true"
    },
    "AXSRFTGenerationHealthIndicator": {
      "enabled": "true"
    },
    "CustomDataSourceHealthIndicator": {
      "enabled": "true"
    },
    "DirServiceConnectivityHealthIndicator": {
      "enabled": "true"
    },
    "ProcessEngineConnectivityHealthIndicator": {
      "enabled": "true"
    },
    "RulesEngineConnectivityHealthIndicator": {
      "enabled": "true"
    },
    "UmConnectionHealthIndicator": {
      "enabled": "true"
    },
    "MyFavIndicator" : {
      "enabled": "true",

```

```
"properties": {
  "myIndicatorProperty": "propValue"
}
}
```

where *MyFavIndicator* is the name of your custom health check class, and *myIndicatorProperty* is a custom configuration property, defined in your code. Add an entry in the *healthCheck.json* file for each indicator property you define in the code.

Example: MyFavIndicatorConfig.java

A sample class that defines the configuration properties of a custom health indicator for My webMethods Server containers. This class is required to map the configuration, supplied through the *healthCheck.json* file and transfer it to the My webMethods Server runtime.

```
package caf.war.CustomHealthIndicatorProj.customhealthindicatorportlet;
import java.util.Map;
import java.util.Optional;
import com.webmethods.rtl.container.health.IHealthIndicatorConfig;
public class MyFavIndicatorConfig implements IHealthIndicatorConfig {
  private boolean enabled;
  private Map<String, Object> properties;
  @Override
  public boolean isEnabled() {
    return this.enabled;
  }
  public void setEnabled(boolean enabled) {
    this.enabled = enabled;
  }
  @Override
  public Map<String, Object> getProperties() {
    return this.properties;
  }
  public void setProperties(Map<String, Object> properties) {
    this.properties = properties;
    // Modify according to your specific data types
    this.myIndicatorProperty = getConfigProperty("myIndicatorProperty", String.class);
    //$NON-NLS-1$
  }
  protected <T extends Object> T getConfigProperty(String key, Class<T> returnType) {
    return
Optional.ofNullable(this.properties.get(key)).map(returnType::cast).orElse(null);
  }
  // Implement for all custom properties, listed in healthCheck.json
  private String myIndicatorProperty;
  // Implement for all custom properties, listed in healthCheck.json
  public String getMyIndicatorProperty() {
    return this.myIndicatorProperty;
  }
}
```

Example: MyFavIndicator.java

A sample class that creates a custom health indicator for My webMethods Server containers.

```

package caf.war.CustomHealthIndicatorProj.customhealthindicatorportlet;
import java.util.HashMap;
import java.util.Map;
import org.osgi.service.component.annotations.Component;
import org.springframework.boot.actuate.health.Health;
import org.springframework.boot.actuate.health.Status;
import com.fasterxml.jackson.databind.JsonNode;
import com.fasterxml.jackson.databind.ObjectMapper;
import com.webmethods.rtl.container.health.IHealthIndicator;
@Component(
    service = {
        IHealthIndicator.class
    },
    property = {
        "name:String=myfavindicator",
        "service.ranking:Integer=1"
    }
)
public class MyFavIndicator implements IHealthIndicator<MyFavIndicatorConfig> {
    private MyFavIndicatorConfig config;
    private Class<MyFavIndicatorConfig> configClass;
    public MyFavIndicator() {
        this.configClass = MyFavIndicatorConfig.class;
    }
    @Override
    public void configure(JsonNode configNode) throws Exception {
        if (configNode != null) {
            this.config = new ObjectMapper().treeToValue(configNode, this.configClass);
        }
    }
    @Override
    public MyFavIndicatorConfig getConfiguration() {
        return this.config;
    }
    // Implement custom indicator properties and their values
    @Override
    public Health health() {
        Map<String, Object> details = new HashMap<>();
        if (config != null) {
            details.put("myProp is", config.getMyIndicatorProperty()); //$NON-NLS-1$
        }
        return Health.status(new
Status(HealthStatus.STATUS_OK.getKey()).withDetail(getName(), details).build();
    }
}

```

Monitoring My webMethods Server Containers with Prometheus

Prometheus is a set of tools for system monitoring that can be configured to monitor a Docker instance, and to collect and analyze metrics for running containers. My webMethods Server containers provide a dedicated endpoint at `http://container_host:port/metrics` which exposes My webMethods Server performance metrics in a Prometheus-compatible format. The supported metric type is gauge. For more information about metrics and metric types, see the Prometheus documentation.

Accessing the My webMethods Server metrics endpoint from the Prometheus server requires basic authentication with an account that is a member of the Admin role.

The My webMethods Server performance metrics provide information various server caches, database connections, loading and execution times for various requests and actions, JVM memory, user sessions, and so on. Help strings in Prometheus-compatible format are available for all metrics at `http://container_host:port/metrics`.

Clustering My webMethods Server Containers

Although you can create a cluster from My webMethods Server containers manually by running the containers against the same database and using the same JMS provider, the recommended approaches to create a clustered environment are using Docker Compose or Kubernetes.

Certain limitations apply when creating and using clusters from My webMethods Server containers:

- Regardless of the selected approach, you cannot create a hybrid cluster, which is a cluster that includes both My webMethods Server instances, installed on premise regular servers/VMs and instances that run in Docker containers. When adding a new cluster node that runs either in a container or from an on-premise installation, My webMethods Server checks whether the node type matches the cluster type, and shuts down the node if the validation fails.
- You cannot run containers with different My webMethods Server versions in the same cluster.
- You cannot restart or shut down the cluster, or individual cluster members from the Cluster Administration page. Placing a node in maintenance mode from the **Cluster Administration** page overrides the maintenance configuration from the image or the `/MWS/volumes/configs/instance_cfg` directory (if available) on container restart.
- When using Kubernetes for clustering, you should define the My webMethods Server using StatefulSet and Ingress objects.

Configuring a My webMethods Server Container Cluster

My webMethods Server containers provide a dedicated location `/MWS/volumes/configs/cluster_cfg` to which you can mount an external volume and supply the initial configuration of a container cluster in a `cluster.xml` file. You can define the external volume in a Docker Compose template or Kubernetes object descriptor. When the cluster starts, My webMethods Server checks whether a volume is mounted to the `/MWS/volumes/configs/cluster_cfg` and copies the `cluster.xml` file from the volume to the database for all cluster nodes to access. Cluster nodes continue to use the `cluster.xml` after cluster restart, and until you supply a new configuration through the volume. You cannot use local `cluster.xml` files when running a My webMethods Server cluster that consists of Docker containers.

You can modify the initial configuration further, using the **Cluster Settings > Advanced or Clustered Configuration > JMS Provider URL > Advanced Settings**, for example, to enable SSL communication to the Universal Messaging server that a cluster node uses. Changes that you make in the My webMethods Server web user interface are stored in the `MWS/volumes/data` directory on the container file system, or the volume mounted to it, and persist after restarting the container. These settings apply per node and override the configuration supplied through the volume, mounted to the `/MWS/volumes/configs` directory if such volume is available.

My webMethods Cluster Sample

Software AG provides the following sample for creating a My webMethods Server cluster with Docker Compose. You can modify the sample according to your use case.

Sample Template for Clustering My webMethods Server Containers

Example: Run My webMethods Server Containers with Docker Compose

The following Docker Compose sample creates a My webMethods Server with two nodes `MWS_node1` and `MWS_node2` from the image `myImageRepo/mws/server:myMwsImage`, and also creates a Universal Messaging container to use as the JMS provider for the cluster. You must configure the My webMethods Server nodes to use the Universal Messaging server instance on the **Cluster Administration** page in My webMethods after starting up the containers. To enable the containers to communicate you must also setup the container networking.

```
version: "3"
services:
  mws_node1:
    image: "myImageRepo/mws/server:myMwsImage"
    container_name: node1
    environment:
      - DB_TYPE=ms
      - DB_USERNAME=webmuser
      - DB_PASSWORD=webmpass
      - DB_URL=jdbc:wm:sqlserver://db.host:port;databaseName=WEBMDB
      - NODE_NAME=MWS_node1
    ports:
      - "8686:8585"
    volumes:
      - ./node1_logs_vol:/opt/softwareag/MWS/volumes/logs
  mws_node2:
    image: "myImageRepo/mws/server:myMwsImage"
    container_name: MWS_node2
    environment:
      - DB_TYPE=ms
      - DB_URL=jdbc:wm:sqlserver:db.host:port;databaseName=WEBMDB
      - DB_USERNAME=webmuser
      - DB_PASSWORD=webmpass
      - NODE_NAME=MWS_node2
    ports:
      - "8585:8585"
    volumes:
      - ./node2_logs_vol:/opt/softwareag/MWS/volumes/logs
  um_server:
    image: myImageRepo/um/um:myUmImage
    container_name: um_server
    ports:
      - "9010:9000"
```


6 Running My webMethods Server from the Command Line

■ Basic Command Line Syntax for My webMethods Server	72
■ Executing My webMethods Server Commands	73
■ Accessing the Command Line Help Contents	73
■ My webMethods Server Instance Administration Commands	74
■ My webMethods Server Instance Operation Commands	79
■ My webMethods Server Instance Configuration Commands	82
■ My webMethods Server Service Management Commands	84
■ My webMethods Server OSGi Profile Commands	86
■ Start, Stop and Execute My webMethods Server Commands on Multiple Server Instances	87
■ Configuring JVM Properties for My webMethods Server Commands	88
■ Log Files for mws Commands	88

Basic Command Line Syntax for My webMethods Server

The following table describes the basic command line syntax for My webMethods Server:

For Windows: `mws.bat -option -option ... command parameter|-Dparameter=value`
For UNIX: `mws.sh -option -option ... command parameter|-Dparameter=value`

where *-parameter* is a native command parameter and *-Dparameter* is a Java system parameter.

You can list more than one option and more than one parameter in a command, using space as separator.

For commands that do not accept Java system parameters, supply the parameters as additional Java properties in the `custom_wrapper.conf` file.

For information about the parameters that each command supports, see the "Parameters" section for the command.

Some My webMethods Server commands do not apply to instances, running in Docker containers. Runtime limitations are listed in the Usage Notes section of the command descriptions.

Options for My webMethods Server Commands

The command line interface for My webMethods Server supports the following options, but each My webMethods Server command accepts only the options listed in the Options section for the command. The following table lists the available command line options:

Option	Description
-p	Full path to the platform installation directory. The default is the <i>Software AG_directory</i> .
-d debug	Starts the server in debug mode. A Java debug listener opens on port 10033, and DEBUG statements appear in the console window.
-n	In a clustered environment, the node name assigned to the server instance. Not required if the server is running standalone.
-s	The name of the server instance. Not required if you are controlling the default instance. The default value is <code>default</code> .
-w	The period of time in seconds to wait for the command to execute.

Executing My webMethods Server Commands

You can perform basic administration and configuration operations using the My webMethods Server command line utility. Starting the server from the command line, for example, allows the use of debug mode so you can record or display server activity.

> To execute My webMethods Server commands

1. At a command line prompt, go to the My webMethods Server `bin` directory:

```
Software AG_directory\MWS\bin
```

2. To run a command, type a My webMethods Server command with the required options and parameters and press **Enter**.

Example

```
mws new -Dserver.name=testInstance -Dhttp.port=8586 -Ddb.type=sqlserver  
-Ddb.url=jdbc:wm:sqlserver://dbserver:1433;databaseName=WMDB  
-Ddb.username=sa -Ddb.password=manage
```

The example command creates a new My webMethods Server instance named `testInstance`, accessible on port 8586. The new server instance uses an SQL Server database with the name `WMDB`, installed on the `dbserver` machine, port 1433. The username and password to connect to the database are `sa` and `manage`.

Accessing the Command Line Help Contents

The command line utility for My webMethods Server provides command line help that you can access in the command prompt, using the `help` command. The `help` command does not accept any options or parameters.

> To get the command line help contents

1. At the command prompt, type the following command to go to the `bin` directory of My webMethods Server:

```
cd Software AG_directory\MWS\bin
```

2. Type the following command:

```
mws help
```

My webMethods Server Instance Administration Commands

The following commands perform My webMethods Server instance administration operations.

new

Creates a new server instance. Specify instance details as command parameters in the following format: `-Dparameter=value`.

Options

The `new` command does not accept any options.

Parameters

The following table lists the parameters that the `new` command accepts.

Parameter	Description
<code>server.name</code>	Optional. Name of the server instance. Specify a unique name among server instances on the machine. If you do not specify a value, the instance name is <code>default</code> .
<code>http.port</code>	Optional. Port number on which the server instance listens. Specify a unique port number among server instances on the machine. The default port number is <code>8585</code> .
<code>install.service</code>	Optional. Whether to install a new My webMethods Server instance as an application or a service. Valid values are: <ul style="list-style-type: none"> ■ <code>true</code> - Install as a service. ■ <code>false</code> - Default. Install as an application.
<code>node.name</code>	Optional. The name of the cluster node that hosts the server instance. If you do not specify a name, the default value is <code>master</code> .
<code>db.type</code>	Required. The type of database used by the server instance. Valid values are: <ul style="list-style-type: none"> ■ <code>ms</code> - Microsoft SQL Server ■ <code>oracle</code> - Oracle ■ <code>db2</code> - DB2 ■ <code>mysql</code> - MySQL Enterprise Edition

Parameter	Description
	<ul style="list-style-type: none"> ■ mysqlce - MySQL Community Edition ■ postgresql - PostgreSQL
db.url	<p>Required. Database connection URL, based on the type of database and the driver.</p> <p>On UNIX systems, you must escape all semicolon characters in the database connection URL with backward slashes, or enclose the whole database URL parameter in double quotes, as follows:</p> <pre style="background-color: #f0f0f0; padding: 5px;">"-Ddb.url=jdbc:wm:database://host:port;databaseName=name"</pre>
db.driver	Optional. The class name of the JDBC driver used to connect to the My webMethods Server database. Required when using a JDBC driver different than the one supplied by Software AG.
db.username	Required for all external databases. User name assigned to the My webMethods Server database.
db.password	Required for all external databases. Password of the My webMethods Server database user.
jndiProviderUrl	Required. The URL of the Universal Messaging server to use as a JMS provider. If you do not specify a JNDI provider URL, or the Universal Messaging server is unavailable, the My webMethods Server instance starts in maintenance mode.
https.port	Optional. The HTTPS listener port. A value of 0 disables the listener.
debug.port	Optional. The Java debug port. The default port number is 10033.
jmx.port	Optional. The JMX port for the new server instance. The default port number is 5002.
http.proxy.host	Optional. The proxy host name.
http.proxy.port	Optional. The proxy port number.
http.proxy.user	Optional. The proxy user name.
http.proxy.password	Optional. The proxy password.
components.include	Optional. Components in the <i>Software AG_directory\MWS\components</i> directory to include in the new server instance.
components.exclude	Optional. Components in the <i>Software AG_directory\MWS\components</i> directory to exclude from the new server instance.

Parameter	Description
<code>components.override</code>	<p>Optional. Specifies whether to overwrite the component files in the <i>Software AG_directory\MWS\server\serverName\deploy</i> directory that are older than the component files in the <i>Software AG_directory\MWS\components</i> directory. Valid values are:</p> <ul style="list-style-type: none"> ■ <code>true</code> - Overwrite component files in the <code>\deploy</code> directory. ■ <code>false</code> - Default. Do not overwrite component files in the <code>\deploy</code> directory.

Usage Notes

- Before creating a new server instance, use the Database Component Configurator to create a unique database or tablespace for the instance, as described in *Installing Software AG Products*. Not required when creating a node in a My webMethods Server cluster.
- After creating a new server instance, initialize the instance using the `init` command. The first initialization of a new My webMethods Server instance takes several minutes to complete. After the first initialization, the server automatically shuts down.
- Not supported when My webMethods Server is running in a container.

Examples

- To create a new instance with name `test`, running on port `8090`, with an external MS SQL Server database with name `my_wm_sql`, installed on the server `db_server`:

```
> mws new -Dserver.name=test -Dhttp.port=8090
-Ddb.type=ms -Ddb.url=jdbc:wm:sqlserver://db_server:1433;
DatabaseName=my_wm_sql;SelectMethod=direct;MaxPooledStatements=100
-Ddb.username=mws_user -Ddb.password=password
[Configuration output displayed in console window...]
> mws -s test run
```

My webMethods Server connects to the database using the username `mws_user`, and the password `password`.

init

Starts My webMethods Server, initializes the My webMethods Server database, and then stops the server. Use this command after creating or updating a My webMethods Server instance.

Options

```
-p
-s
-d
```

-n

Parameters

Accepts Java system properties as parameters in the following format:

```
-Dany.java.system.property=value
```

Usage Notes

- Not supported when My webMethods Server is running in a container.

update

Updates classpaths and deploys new or updated components.

Options

-p

-s

-w

Parameters

The following table lists the parameters that the update command accepts.

Parameter	Description
node.name	Optional. Specify a new name for the cluster node that hosts the My webMethods Server instance.
db.type	Optional. The type of database used by the server instance. Valid values are: <ul style="list-style-type: none"> ■ ms - Microsoft SQL Server ■ oracle - Oracle ■ db2 - DB2 ■ mysqlce - MySQL Enterprise Edition ■ mysqlce - MySQL Community Edition
db.url	Optional. The database connection URL. On Unix systems, enclose the database URL parameter in double quotes, as follows: <pre>"-Ddb.url=jdbc:wm:database://host:port;databaseName=name"</pre>

Parameter	Description
<code>db.driver</code>	Optional. The class name of the JDBC driver to use when connecting to the database. Required when using a JDBC driver different than the one supplied by Software AG
<code>db.user</code>	Required for all external databases. User name of the database user.
<code>db.password</code>	Required for all external databases. Password of the database user.
<code>components.override</code>	Optional. Specifies whether to overwrite the component files in the <code>Software AG_directory\MWS\server\serverName \deploy</code> directory that are older than the component files in the <code>Software AG_directory\MWS\components</code> directory. Valid values are: <ul style="list-style-type: none">■ <code>true</code> - overwrite component files in the <code>deploy</code> directory of the server.■ <code>false</code> - do not overwrite component files in the <code>deploy</code> directory. This is the default value.
<code>dont.update.classpath</code>	Optional. Specifies whether to update the generated My webMethods Server classpath. Valid values are: <ul style="list-style-type: none">■ <code>true</code> - do not update the server classpath.■ <code>false</code> - update the server classpath. This is the default value.

Usage Notes

- Not supported when My webMethods Server is running in a container.

delete

Deletes an existing instance of My webMethods Server.

Options

- p
- s
- w

Parameters

The `delete` command does not accept any parameters.

Usage Notes

- The `delete` command does not prompt for confirmation before deleting the instance. Use the command with extreme caution.
- Not supported when My webMethods Server is running in a container.

My webMethods Server Instance Operation Commands

The following commands perform My webMethods Server instance management operations.

run

Starts the server in the same console window.

Options

-p
-s
-n
-d

Parameters

The `stop` command accepts Java system properties as parameters in the following format:

```
-Dany.java.system.property=value
```

Usage Notes

- Not supported when My webMethods Server is running in a container.

start

Starts the server in a new console window. On UNIX based operating systems the process is started in background execution mode.

Options

-p
-s

-n

-d

-w

Parameters

The `start` command accepts Java system properties as parameters in the following format:

```
-Dany.java.system.property=value
```

Usage Notes

- Not supported when My webMethods Server is running in a container.

restart

Stops a running server and then starts it again.

Options

-p

-s

-n

-d

-w

Parameters

The `restart` command accepts Java system properties as parameters in the following format:

```
-Dany.java.system.property=value
```

Usage Notes

- Not supported when My webMethods Server is running in a container.

stop

Stops a running server.

Options

-p

-s

-n

-w

Parameters

The stop command accepts Java system properties as parameters in the following format:

```
-Dany.java.system.property=value
```

Usage Notes

- Not supported when My webMethods Server is running in a container.

ping

Returns information about a running server instance, for example, server ports, roles, front end URL and initialization time. Returns `Portal ping failed.` if the server is stopped. The ping command requires authentication to execute.

Options

-s

-n

-w

Parameters

The following table lists the parameters that the ping command accepts.

Parameter	Description
<code>mws_username</code>	Required. The user name of the administrative My webMethods Server user to authenticate the execution of the command.
<code>mws_password</code>	Required. The password of the administrative My webMethods Server user to authenticate the execution of the command.

Examples

To ping the server instance with name test, using the Administrator user with password manage:

```
mws -s test ping Administrator manage
```

updatesinfo

Displays information about any fixes that have been installed to My webMethods Server, the My webMethods Server Common Library, or the user interfaces of installed webMethods applications.

Options

-s
-n

Parameters

The updatesinfo command does not accept any parameters.

My webMethods Server Instance Configuration Commands

The following commands perform My webMethods Server instance configuration operations.

getconfig

Downloads a configuration file from the My webMethods Server database. Specify the name of the configuration file as a parameter of the command, as follows:

```
mws getconfig fileName
```

For more information and a list of configuration files, stored in the My webMethods Server database, see [“Modifying Configuration Files Stored in the Database” on page 100](#).

putconfig

Uploads a configuration file to the My webMethods Server database. Specify the name of the configuration file as a parameter of the command, as follows:

```
mws getconfig fileName
```

For more information and a list of configuration files, stored in the My webMethods Server database, see [“Modifying Configuration Files Stored in the Database” on page 100](#).

export

Exports My webMethods Server runtime data for migration.

Options

-s

Parameters

The following table lists the parameters that the export command accepts.

Parameter	Description
descriptor_filefull_path	Required. Full path to the descriptor file to use for exporting migration data.
target_folder	Required. Full path to the folder to use to export runtime data.
mws_username	Required. The user name of the administrative My webMethods Server user to authenticate the execution of the command.
mws_password	Required. The password of the administrative My webMethods Server user to authenticate the execution of the command.

import

Imports runtime data from an existing My webMethods Server installation.

Options

-s

Parameters

The following table lists the parameters that the import command accepts.

Parameter	Description
info_properties_filefull_path	Required. Full path to the .properties file to import.
mws_username	Required. The user name of the administrative My webMethods Server user to authenticate the execution of the command.
mws_password	Required. The password of the administrative My webMethods Server user to authenticate the execution of the command.

My webMethods Server Service Management Commands

The following commands administer and operate server instances when installing or using My webMethods Server as a Windows service, or UNIX daemon.

installservice

The `installservice` command registers My webMethods Server as a service on Windows, or a daemon on UNIX systems.

Options

- p
- s
- n
- d

Parameters

On Windows systems, the `installservice` command accepts Java system properties as parameters in the following format:

```
-Dany.java.system.property=value
```

On UNIX systems, you must specify the user name of the user that installs the service.

Usage Notes

- Not supported when My webMethods Server is running in a container.

uninstallservice

The `uninstallservice` command unregisters My webMethods Server as a Windows service or UNIX daemon.

Options

- p
- s

Parameters

The command does not accept additional parameters.

Usage Notes

- Not supported when My webMethods Server is running in a container.

startservice

Applies only to Windows. The `startservice` command starts a My webMethods Server instance, installed as a Windows service.

Options

-p

-s

Parameters

The `startservice` command does not accept any parameters.

Usage Notes

- Not supported when My webMethods Server is running in a container.

stopservice

Applies only to Windows. Stops a My webMethods Server instance, installed as a Windows service.

Options

-p

-s

Parameters

The `stopservice` command does not accept any parameters.

Usage Notes

- Not supported when My webMethods Server is running in a container.

restartservice

Applies only to Windows. Stops and then starts a My webMethods Server instance, installed as a Windows service.

Options

- p
- s

Parameters

The `restartservice` command does not accept any parameters.

Usage Notes

- Not supported when My webMethods Server is running in a container.

My webMethods Server OSGi Profile Commands

The following commands manage the OSGi profiles for My webMethods Server instances.

update-osgi-profile

Updates the OSGi profile for the specified server instance. Resets the server name, the service name, and the `JAVA_HOME` variable, and updates the OSGi bundles associated with the server instance.

Options

- p
- s

Parameters

The `update-osgi-profile` command does not accept any parameters.

Usage Notes

- Use the `update-osgi-profile` command when you modify the configuration of a server instance or install a new webMethods application to run on an existing server instance.
- Before updating an OSGi profile, make a backup copy of the `custom_wrapper.conf` configuration file.

- Not supported when My webMethods Server is running in a container.

Start, Stop and Execute My webMethods Server Commands on Multiple Server Instances

If you want to start, stop, or execute a My webMethods Server command on all server instances at one go, there are commands associated with all server instances, installed on a machine.

You cannot execute My webMethods Server commands on multiple server instances when the instances are running in containers. By default, these scripts are excluded from the official My webMethods Server images and Docker scripts and do not take effect even when explicitly included in a custom image.

➤ To start, stop, or execute a command on all server instances

1. At a command line prompt, type the following command to move to the command's home directory:

```
cd Software AG_directory\MWS\bin
```

2. Type one the following commands:

The following table lists command syntax for executing My webMethods Server commands on multiple server instances:

Purpose	Operating system	Command
Execute a command on all server instances. The following commands are not supported: run new ant	Windows	mwsall.bat
	UNIX	mwsall.sh
	Windows	startall.bat
	UNIX	startall.sh
Start all server instances. The server instances are started consequently in alphabetical order. On Windows operating systems, the command starts a server instance as a service if it has a registered service, otherwise it starts the instance as a process. On UNIX based operating systems, the command starts the instances as processes only.	Windows	startall.bat
	UNIX	startall.sh
Stop the server instances. The command calls mws.{bat sh} stopfor all server instances.	Windows	stopall.bat
	UNIX	stopall.sh

Configuring JVM Properties for My webMethods Server Commands

The command line utility for My webMethods Server operates in a separate JVM, different than the main server runtime environment. When executing My webMethods Server a command, the utility uses its default settings, regardless of the JVM configuration, specified for My webMethods Server in the `custom_wrapper.conf` file. You can customize the JVM setting for the My webMethods Server command utility by supplying a `server.properties` file with the required configuration in the `Software AG_directory\MWS\server\serverName\config\` directory of your server instance.

Example: server.properties Configuration File

```
# JVM configuration
jvm.arg=-Xms32m
jvm.arg=-Xmx1024m
jvm.arg=-server

# SSL options
java.option=-Djavax.net.ssl.keyStore="{server.home}/config/security/localhost.p12"
java.option=-Djavax.net.ssl.keyStorePassword={DES}vrFIelCdkow=
java.option=-Djavax.net.ssl.keyStoreType=pkcs12

java.option=-Djavax.net.ssl.trustStore="{server.home}/config/security/sagdemoca.jks"
java.option=-Djavax.net.ssl.trustStorePassword={DES}vrFIelCdkow=
java.option=-Djavax.net.ssl.trustStoreType=jks
# Proxy configuration
java.option=-DproxySet=false
java.option=-Dhttp.proxyHost=
java.option=-Dhttp.proxyPort=
java.option=-Dhttp.proxyUser=
java.option=-Dhttp.proxyPassword=
java.option=-Dhttp.nonProxyHosts=
```

Supply JVM configuration options as follows:

```
jvm.arg=jvm.configuration.argument
```

Supply Java system properties as follows:

```
java.option=-Dany.java.system.property=value
```

Log Files for mws Commands

For some of the `mws` commands, My webMethods Server creates a log file with information from the command execution in addition to logging the command output to the console. The log file name takes this form:

```
command-name_server-name.log
```

where:

- `command-name` is the name of the `mws` command.

- *server-name* is the name of the server instance affected by the command.

For example, if you issue this command:

```
mws -s alpha create-osgi-profile
```

The resulting log file has the name `create-osgi-profile_alpha.log`.

The log files for `mws` commands are written to this location:

Software AG_directory \MWS\bin

If the `mws` command is unsuccessful, one quick way to check the log file is to search for the existence of this exit code:

```
ExitCode: 13
```

Following that code, you should find the missing requirement, possibly a missing bundle or feature ID. If you cannot find an obvious reason for the failure, be prepared to make a copy of the log file available to Software AG Global Support.

When My webMethods Server is running inside a container, only certain actions, such as applying a test patch or a third-party library at the time of container startup can result in writing a log file to the `bin` directory for the instance. If you want to keep these log files, and you have not specified a volume for log files, create a backup copy before shutting down the container.

7 Modifying Configuration Files

■ The Java Service Wrapper	92
■ Configuring JVM Settings for My webMethods Server	95
■ Modifying Configuration Files Stored in the Database	100
■ Configuring My webMethods Server to Run in 32-bit on Solaris, HP-UX, or Linux	102
■ Configuring HTTP Listeners to Use a Single IP Address	103
■ Configuring Whether Diagnostics Are Executed at Startup	105

The Java Service Wrapper

My webMethods Server runs on the Software AG Common Platform, which in turn runs in a Java Virtual Machine (JVM). The Java Service Wrapper is an application developed by Tanuki Software, Ltd. It is a utility program that launches the JVM in which My webMethods Server runs.

In addition to launching the JVM, the Java Service Wrapper offers features for monitoring the JVM, logging console output, and generating thread dumps. For an overview of the Java Service Wrapper, see *Software AG Infrastructure Administrator's Guide*.

The Java Service Wrapper Configuration Files

For My webMethods Server, the configuration files for the Java Service Wrapper reside in the following directory:

Software AG_directory \profiles\MWS_serverName\configuration

When you start My webMethods Server, the properties in the wrapper configuration files determine the JVM settings and the behavior of the logging and monitoring features of the Java Service Wrapper.

The following table describes the properties that each file contains:

File name	Description
wrapper.conf	Contains property settings that are installed by My webMethods Server. Do not modify the contents of this file unless asked to do so by Software AG.
custom_wrapper.conf	Contains properties that modify the installed settings in wrapper.conf. If you need to modify the property settings for the Java Service Wrapper, you make your changes in this file.
wrapper.conf.template	Contains settings that are applied to the wrapper.conf file when My webMethods Server is updated or upgraded. Do not modify this file. The file is located here: <i>Software AG_directory</i> \MWS\server\serverName\config\

JVM Configuration

When the Java Service Wrapper launches the JVM, it provides configuration settings that, among other things, specify the size of the Java heap, and the directories in the classpath.

The JVM Configuration Properties

The `wrapper.java` properties in the Java Service Wrapper configuration files determine the configuration of the JVM in which My webMethods Server runs.

The JVM property settings that My webMethods Server installs are suitable for most environments. However, you can modify these properties if the installed settings do not suit your needs. For procedures and additional information, see [“Configuring JVM Settings for My webMethods Server” on page 95](#).

The Wrapper Log

My webMethods Server has its own logging mechanism, described in [“Controlling Server Logging” on page 306](#). In addition, the Java Service Wrapper records console output in a log file. The log contains the output sent to the console by the wrapper itself and by the JVM in which My webMethods Server runs. The wrapper log is especially useful when you run My webMethods Server as a Windows service, because console output is normally not available to you in this mode.

The Java Service Wrapper log for My webMethods Server is located in the following file:

```
Software AG_directory \profiles\MWS_serverName\logs\wrapper.log
```

To view the log, open the log file in a text editor.

The Logging Properties

The `wrapper.console` and `wrapper.log` properties in the wrapper configuration files determine the content, format, and behavior of the wrapper log.

The logging settings that My webMethods Server installs are suitable for most environments. However, you can modify the following properties if the installed settings do not suit your needs. For procedures and additional information, see *Software AG Infrastructure Administrator’s Guide*.

The following table lists the available logging properties:

Property	Value
<code>wrapper.logfile.maxsize</code>	Maximum size to which the log can grow.
<code>wrapper.logfile.maxfiles</code>	Number of old logs to maintain.

Fault Monitoring

The Java Service Wrapper can monitor the JVM for the specified conditions and then restart the JVM or perform other actions when it detects these conditions.

The following table describes the fault-monitoring features My webMethods Server uses or allows you to configure.

Feature	Enabled?	User configurable?
JVM timeout	Yes	Yes. See the <code>wrapper.ping</code> properties in “Configuring My webMethods Server Settings” on page 95.
Deadlock detection	No	Yes. See the <code>wrapper.check.deadlock</code> properties in “Configuring Wrapper JVM Checks” on page 96.
Console filtering	Yes	Yes. See “Console Filtering Properties” on page 94.

For more information about fault-monitoring options, see *Software AG Infrastructure Administrator’s Guide*.

Console Filtering Properties

The `wrapper.filter` properties in the wrapper configuration files determine whether the wrapper monitors the console for specified messages. My webMethods Server installs pre-defined filters that monitor the console for certain messages. For information about the installed filters, see [“Configuring JVM Out-of-Memory Checks”](#) on page 97.

You can add additional filters to the Java Service Wrapper using the following properties. For procedures and additional information, see *Software AG Infrastructure Administrator’s Guide*.

The following table lists the custom JVM properties that configure console filtering.

Property	Value
<code>wrapper.filter.trigger.n</code>	String of text that you want to detect in the console output.
<code>wrapper.filter.action.n</code>	Action that occurs when the Java Service Wrapper detects the string of text.
<code>wrapper.filter.allow_wildcards.n</code>	Flag (TRUE or FALSE) that specifies whether the Java Service Wrapper processes wildcard characters that appear in <code>wrapper.filter.trigger.n</code> .
<code>wrapper.filter.message.n</code>	Message that displays when the Java Service Wrapper detects the string of text.

Generating a Thread Dump

The Java Service Wrapper provides a utility for generating a thread dump when My webMethods Server is running as a Windows service. A thread dump can help you locate thread contention issues that can cause thread blocks or deadlocks.

For information about generating a thread dump using the Java Wrapper Service, see *Software AG Infrastructure Administrator’s Guide*.

Configuring JVM Settings for My webMethods Server

You can modify JVM settings for a My webMethods Server instance in the `custom_wrapper.conf` file.

> To modify JVM settings

1. Open the `custom_wrapper.conf` file for the server instance in a text editor. You can find the file at this location:

```
Software AG_directory \profiles\MWS_serverName\configuration\
```

2. Add or modify statements as needed and save the file.
3. Restart My webMethods Server.

Configuring My webMethods Server Settings

You can use the `custom_wrapper.conf` file to configure the ping timeout for a My webMethods Server instance. The following table lists the custom JVM properties to add in the `custom_wrapper.conf` file:

Parameter	Description
<code>wrapper.ping.timeout=value</code>	Integer. The amount of time in seconds to wait between two ping responses from the JVM. The default value is 120. If set to 0, the wrapper will never time out, but a real indicator whether the server is suspended will not exist either.

Setting Initial and Maximum Memory Limits

You can configure the initial and maximum amount of memory that is allocated by the JVM at startup. The following table lists the initial and maximum memo limits, set by default by the Java Service Wrapper:

Parameter	Value in megabytes
<code>wrapper.java.initmemory=</code>	128
<code>wrapper.java.maxmemory=</code>	1024

To use the default values that are configured in the JVM itself, specify zero (0) in the properties of the `custom_wrapper.conf` file. For example:

```
wrapper.java.maxmemory=0
```

Note:

If you specify a value for the `wrapper.java.initmemory` memory property, make sure that it is smaller than the value of the `wrapper.java.maxmemory` property.

Raising the Maximum Memory Limit on UNIX

On UNIX or Linux systems, if My webMethods Server fails to initialize, you may need to increase the maximum amount of available memory or the maximum open files limit. This need can occur if you have multiple language packs installed, or if there are multiple webMethods components installed on the server.

In the `custom_wrapper.conf` file, specify the maximum memory limit in megabytes. To double the maximum available memory, for example, change this value:

```
wrapper.java.maxmemory=1024
```

to this:

```
wrapper.java.maxmemory=2048
```

Each UNIX platform has its own method for configuring Open File limits. Here is an example on how to increase the maximum Open Files limit parameters on Linux:

```
ulimit -c unlimited
ulimit -n 8192
```

Check with your administrator to make these changes on Linux or other UNIX platforms.

For information on editing this `custom_wrapper.conf`, see [“Configuring JVM Settings for My webMethods Server” on page 95](#).

Configuring Wrapper JVM Checks

The Java Service Wrapper can perform internal detection of deadlock threads or out of memory conditions within the JVM. By default, the deadlock detection is disabled in the My webMethods Server service and if you want to use this functionality, you need to enable it manually in the `custom_wrapper.conf` file.

The following table describes the additional parameters for configuring JVM checks:

Parameter	Description
<code>wrapper.check.deadlock=true false</code>	Boolean. Enables or disables JVM deadlock checks. The default value is <code>FALSE</code> - deadlock checking is disabled.
<code>wrapper.check.deadlock.interval=interval</code>	Integer. The deadlock detection interval in seconds. The default value is <code>60</code> .

Parameter	Description
<code>wrapper.check.deadlock.action=action</code>	String. The action to perform when a deadlock occurs. The default value is RESTART.
<code>wrapper.check.deadlock.output=output</code>	String. The output of the wrapper. Valid values are: <ul style="list-style-type: none"> ■ FULL ■ SIMPLE ■ NONE <p>The default value is FULL.</p>

Configuring JVM Out-of-Memory Checks

You can configure the properties that control and resolve the out-of-memory conditions by filtering the console output of the JVM. These properties use this syntax:

```
wrapper.filter.element.<n>=parameter
```

The number *n* starts from one and increases by one for each consecutive element. As a rule, the chain of additional parameters must start at 1 and be consecutive. However, if you enable `wrapper.ignore_sequence_gaps` property, the sequence can be broken.

The following table lists the available parameters, their recommended values, and the corresponding descriptions.

Parameter	Recommended Value	Description
<code>wrapper.filter.trigger.n=</code>	<code>java.lang.OutOfMemory Error</code>	Enables a query for an out-of-memory condition on the console output of the JVM.
<code>wrapper.filter.action.n=</code>	RESTART	Controls the action if the configured filter is met (an out-of-memory condition occurs).
<code>wrapper.filter.message.n=</code>	The JVM has run out of memory.	Controls the message that is displayed if the filter is met (an out-of-memory condition occurs).

Additional JVM Parameters

There are some parameters that do not relate to My webMethods Server but to the JVM itself. You set custom JVM parameters in the `custom_wrapper.conf` file for My webMethods Server, using the following syntax:

```
wrapper.java.additional.n=parameter
```

The number *n* starts from one and increases by one for each additional element. As a rule, the chain of additional parameters must start at one and be consecutive. For example:

```
# Java Additional Parameters
wrapper.java.additional.1=-Dosgi.install.area="%OSGI_INSTALL_AREA%"
wrapper.java.additional.2=-Declipse.ignoreApp=true
wrapper.java.additional.3=-Dosgi.noShutdown=true
```

The following table describes some of the additional parameters, available for customizing JVM settings:

Parameter	Description
<code>wrapper.java.additional.n=-DproxySet=true</code>	Starts the program without a graphical user interface window.
<code>wrapper.java.additional.n=-Dhttp.proxyHost=host_name</code>	The host name of the proxy server.
<code>wrapper.java.additional.n=-Dhttp.proxyPort=port_number</code>	The port number on the proxy server.
<code>wrapper.java.additional.n=-Dhttp.proxyUser=user_name</code>	A user name used for authentication.
<code>wrapper.java.additional.n=-Dhttp.proxyPassword=password</code>	A password used for authentication.
<code>wrapper.java.additional.n=-Dhttp.nonProxyHosts=bypass_hosts</code>	A list of hosts that should be reached directly, bypassing the proxy. Entries are separated by a vertical bar ().

Configuring the Axis HTTP Client

Making multiple concurrent SOAP requests from a portlet in My webMethods Server to a web service in Integration Server might result in longer response times, or request timeouts. In such cases, you can modify the connection settings of the built-in Axis client for My webMethods Server, by including additional parameters in the `custom_wrapper.conf` file.

The following table lists the custom JVM parameters that you use to configure the Axis client in My webMethods Server.

Parameter	Description
<code>axisclient.DefaultMaxConnectionsPerHost</code>	The maximum number of concurrent connections to a service on a host. The default value is 10.
<code>axisclient.MaxTotalConnections</code>	The maximum number of active connections for a single client instance. The default value is 100.

For more information about modifying the `custom_wrapper.conf` file, see [“The Java Service Wrapper” on page 92](#).

Configuring Optimized Startup for On-premises Installations

When starting up in a container, My webMethods Server uses an optimization mechanism to reduce startup time. The optimization mechanism launches an in-memory instance of the Derby database to bootstrap My webMethods Server, then transfers the data to the production database, avoiding most of the network latency overhead. By default, on-premises My webMethods Server installations do not use optimized start mechanism, but you can switch it on by adding the following JVM property to the `custom_wrapper.conf` file:

Parameter	Description
<code>mws.inmemory.init</code>	Boolean. Whether to use the database query optimization mechanism during start up. The default value is <code>false</code> . Set to <code>true</code> , to enable optimized startup.

Changing the Truststore for OAuth Authentication

By default, the OAuth2 authentication handler for My webMethods Server uses a built-in dedicated truststore to load security certificates for OAuth2 authentication. To utilize certificates from the common My webMethods Server truststore you must disable the use of the dedicated truststore by adding the following JVM property to the `custom_wrapper.conf` file:

Parameter	Description
<code>use.oauth.client.truststore</code>	Boolean. Whether to use the default (built-in) truststore of the OAuth handler. The default value is <code>true</code> . Set to <code>false</code> to disable the use of the dedicated OAuth trust store and use the common My webMethods Server truststore instead.

Modifying Configuration Files Stored in the Database

Most of the server configuration files are stored in the My webMethods Server database by default, instead of in the local file system. To edit any My webMethods Server configuration file, you must download the file from the database, using the `mws getconfig` command. When you modify a configuration file, the My webMethods Server instance can be either running or stopped, but the database must be running.

For more information about the configuration files that you can modify, see [“Configuration Files Stored in the Database”](#) on page 101.

➤ To edit configuration files for My webMethods Server

1. At a command line prompt, type the following command to move to the server’s bin directory:

```
cd Software AG_directory\MWS\bin
```

2. To retrieve a configuration file from the My webMethods Server database, type this command:

```
mws getconfig fileName
```

where *fileName* is the name of the configuration file that you want to edit.

For example:

```
mws getconfig cluster.xml
```

```
mws getconfig logging.properties
```

3. Open the downloaded configuration file in a text editor and modify it as needed.

You can find the file at this location:

```
Software AG_directory \MWS\server\serverName\config
```

4. To deploy the revised file to the My webMethods Server database, type this command:

```
mws putconfig fileName
```

5. Delete the file from the `\serverName\config` directory.

If you do not delete the file, the server instance will continue to use the local version of the configuration file.

6. Restart the node using this command:

```
mws -s serverName restart
```

Changes to configuration files are not applied until after a restart.

Configuration Files Stored in the Database

The following table shows the files you can download using the `mws getconfig` command and their purpose.

Configuration file	Purpose
cache.xml	Increase cache size.
cluster.xml	Supply and store configuration data for each node in a cluster.
defaultPortletAppWeb.xml	Modify properties of a web application.
defaultPartitionPortlets.properties	List the portlets affected by the phase provider for the default cluster partition. If you have multiple cluster partitions, it is possible to have multiple portlets properties files, each having the file name <i>partitionNamePortletsProperties.xml</i> .
email.properties	Configure e-mail servers.
iconMap.properties	Map mime types (Content types) to icons located in the <code>ui\images</code> directory.
jetty.xml	Configure the Jetty server.
log4j.init.properties	Turn internal debugging on or off (Apache Log4J logging package).
log4j.override.properties	Modify properties of the Apache Log4J logging package.
logging.properties	Customize logging folders and patterns.
mimeTypes.properties	Customize mime types.
perfUtil.properties	Enable and disable the performance service.
phaseProvider.xml	Modify the server polling interval. If you have multiple cluster partitions, it is possible to have multiple phase provider files, each having the file name <i>partitionNamePhaseProvider.xml</i> .
storageConfig.xml	Configure content services.
systemPaths.properties	Configure the location of logs or the temp folder.
systemProperties.properties	Store persisted system properties.
defaultPortletAppWeb.xml	Modify the properties of a web application.
email.properties	Configure email servers.

Configuration file	Purpose
websso.properties	Configure single sign-on for a third-party identity provider (IDP) using Security Assertion Markup Language (SAML).

Note:

File names and file contents are case sensitive.

Configuring My webMethods Server to Run in 32-bit on Solaris, HP-UX, or Linux

On Solaris, HP-UX, or Linux systems, it is necessary to specify whether the computer has a 32- or 64-bit architecture. By default, My webMethods Server is configured to run in 64-bit mode. You can switch the server to 32-bit mode.

➤ **To configure My webMethods Server to run in 32-bit mode on Solaris or HP-UX**

1. Use this command to stop My webMethods Server:

```
mws -option stop
```

Options are described in [“Basic Command Line Syntax for My webMethods Server”](#) on page 72.

2. Open this file in a text editor:

```
Software AG_directory /MWS/bin/setenv.sh
```

3. Locate this text:

```
rem to use 64 bit JVM uncomment next line  
set JAVA_D64=-d64
```

and comment the second line:

```
rem to use 64 bit JVM uncomment next line  
rem set JAVA_D64=-d64
```

4. Save and close the file.
5. Use this command to update the OSGi profile:

```
mws -option update-osgi-profile
```

6. Restart My webMethods Server.

Configuring HTTP Listeners to Use a Single IP Address

My webMethods Server binds its listener ports on all available IP addresses. When you have installed My webMethods Server on server hardware with multiple network interfaces, you can bind the HTTP listener ports to a single IP address.

Configuring Jetty Listeners to Use IP Addresses

Jetty creates several listening ports on the My webMethods Server.

- Ports for HTTP and HTTPS
- Random socket acceptor ports for HTTP and HTTPS

If you have installed My webMethods Server on server hardware with multiple network interfaces, you can bind the HTTP listener ports to a single IP address. Use this procedure to configure a specific IP address to the port used by the Jetty listeners.

Configuring the Jetty listening ports requires specifying the IP addresses in the `custom_wrapper.conf` configuration file.

➤ To configure specific IP addresses to be used by Jetty listeners

1. Open the `custom_wrapper.conf` configuration file for the server instance in a text editor.
2. Add the following JVM property:

```
wrapper.java.additional.nnn=-Djetty.host=IP_address
```

where *nnn* is the consecutive number of the JVM property, and *IP_address* is the IP address to bind the HTTP listener ports.

3. Save the file and restart My webMethods Server.

Configuring an IP Address for the JCR Repository RMI Server

When using a Java Content Repository (JCR), the connection to the JCR is implemented using an RMI to communicate with My webMethods Server. The default JCR listener port is 10999.

If you have installed My webMethods Server on server hardware with multiple network interfaces, you can bind the HTTP listener ports to a single IP address.

In the `custom_wrapper.conf` file, specify a value for the following parameter:

```
wrapper.java.additional.204=-Dmws.jcr.rmi.bind.address=IP_address|host_name
```

where *IP_address* or *host_name* is the IP address or host name of the JCR repository RMI server listener.

For information on editing this file, see [“Configuring JVM Settings for My webMethods Server” on page 95](#).

Configuring IP Addresses in GLUE Web Services Registry

If your instance of My webMethods Server has Central Configuration installed, the GLUE web services registry was created to use a range of ports from 16000 to 16025. You configure the IP address in the `GlueServiceRegistryProperties.xml` file to use a specific IP address.

If you have installed My webMethods Server on server hardware with multiple network interfaces, you can bind the HTTP listener ports to a single IP address. Use this procedure to configure a specific IP address to the port used by the GLUE web services registry.

➤ To configure a specific IP address

1. Go the installation directory for My webMethods Server, navigate to `MWS/server/default/config/engine`, and open the `GlueServiceRegistryProperties.xml` file in an editor.

2. Look for the following property:

```
<entry key="service.address">0.0.0.0</entry>
```

3. Replace the value, `0.0.0.0`, with the IP address.

Configuring My webMethods Server on Multi-Home Machines

If you are configuring My webMethods Server on multi-home machines, that is machines that have multiple network interfaces and IP addresses, use the following procedure to bind My webMethods Server listeners to a single IP address.

➤ To configure My webMethods Server on multi-home machines

1. Shut down My webMethods Server.

2. Bind the HTTP and HTTPS listeners to the IP address as described below. Perform the steps for each server instance.

- a. Go to the *Software AG_directory* `/profiles/MWS_instanceName/configuration` directory, and open the `custom_wrapper.conf` file in a text editor.

- b. Add the following custom JVM property:

```
wrapper.java.additional.nnn=-Djetty.host=IP_address
```

where *nnn* is the consecutive number of the JVM property, and *IP_address* is the IP address to bind the HTTP listener ports.

- c. Save the `custom_wrapper.conf` file.
3. For each cluster node, go to the *Software AG_directory* /MWS/server/*serverName*/config/engine directory, open the `GlueServiceRegistryProperties.xml` file, and specify the IP address on the following element:

```
<entry key="service.address">IP_address</entry>
```

4. For each cluster node, bind the SOAP monitor and JCR servers to the IP address as follows:
 - a. Go to the *Software AG_directory* /profiles/MWS_*serverName*/configuration directory and open the `wrapper.conf` and `custom_wrapper.conf` files.
 - b. Specify values for the following parameters:

```
wrapper.java.additional.203=-Dsoap.monitor.bind.address=  
IP_address|host_name
```

where *IP_address* or *host_name* is the IP address or host name of the SOAP Monitor portlet listener.

```
wrapper.java.additional.204=-Dmws.jcr.rmi.bind.address=  
IP_address|host_name
```

where *IP_address* or *host_name* is the IP address or host name of the JCR repository RMI server listener.

5. The JMX server will not use the `wrapper.conf` directive to bind to an IP address, so you must specify the binding in the JMX connector properties file instead. For each cluster node, do the following:
 - a. Go to the *Software AG_directory* /profiles/MWS/*serverName*/configuration/com.softwareag.platform.config.propsloader directory and back up the `com.softwareag.jmx.connector.pid-500number.properties` file.
 - b. Open the file and add the following line:

```
host=IP_address
```

6. Restart My webMethods Server.

Configuring Whether Diagnostics Are Executed at Startup

By default at startup, My webMethods Server performs the following diagnostic actions:

- Determines the fixes that have been applied to My webMethods Server
- Executes the dbintegritycheck tool that searches for potential database errors

You can prevent My webMethods Server from performing one or both of these diagnostic actions by updating the `phaseProvider.xml` configuration file. The `phaseProvider.xml` configuration file is located in the database or under `My webMethods Server_directory \server \serverName \config`.

Preventing My webMethods Server from Listing Installed Fixes at Startup

My webMethods Server determines the installed fixes at server startup and logs the list of installed fixes to the `_full_.log` file.

Because determining the list of fixes might take a few minutes, you might want to disable this action at server startup. To prevent My webMethods Server from listing installed fixes at startup, edit the `startupDiagnostics` section of the `phaseProvider.xml` configuration file as shown below.

```
<Phase class="com.webmethods.portal.system.init.impl.DefaultPhase"
      enabled="true" name="startupDiagnostics">
  <PhaseInfo class="com.webmethods.portal.system.impl.UpdatesInfo"
            enabled="false" name="updatesInfo"/>
  .
  .
  .
</Phase>
```

Preventing My webMethods Server from Executing dbintegritycheck at Startup

The My webMethods Server automatically executes the `dbintegritycheck` tool at startup and logs the results to the `_full_.log` file. The `dbintegritycheck` tool reviews the dynamic business objects (DBOs) deployed to My webMethods Server to search for potential errors that might have occurred if a DBO was installed or upgraded incorrectly.

Although it is recommended that you keep this action enabled, you can prevent My webMethods Server from executing `dbintegritycheck` at startup. To do so, edit the `startupDiagnostics` section of the `phaseProvider.xml` configuration file as shown below.

```
<Phase class="com.webmethods.portal.system.init.impl.DefaultPhase"
      enabled="true" name="startupDiagnostics">
  <PhaseInfo
    class="com.webmethods.portal.system.impl.DbIntegrityCheckLauncher"
    enabled="false" name="dbIntegrityCheck"/>
  .
  .
  .
</Phase>
```

II My webMethods Administrator Functions

8	Managing My webMethods Configuration	109
9	Searches for Users, Groups, and Roles	141
10	Managing Users and Groups	149
11	Managing Permissions	169
12	Managing Roles and Access to My webMethods	183
13	My webMethods Server Clustering	199

8 Managing My webMethods Configuration

■ Managing Directory Services	110
■ Managing External Directory Services	112
■ Managing External Data Sources	129
■ Managing Email Settings	137
■ My webMethods Server and Multi-Factor Authentication	138
■ Managing Calendars	140

Managing Directory Services

A *directory* is similar to a database in that it contains a collection of entries (in this case, individuals), each of which has a set of attributes, such as names, email addresses, and so forth. A *directory service* provides a mechanism for delivering information about the entries in the directory.

My webMethods Server includes an internal directory service. However, if you are using an external directory (for example, if you are using Lightweight Directory Access Protocol (LDAP)), you can configure My webMethods Server to also access user and group information from the external directory service.

Internal Directory Service

The internal directory service that is provided by default with your My webMethods Server installation is called the *system directory service*. My webMethods Server stores information about users, groups, and roles that you define in this system directory service to the My webMethods Server database. Use the system directory service if you need to maintain only a moderate number of users and groups.

The following table lists the users and groups, defined by default in the system directory service:

User Name	Description
My webMethods Administrator	<p>The default administrator of My webMethods. This user can perform user management functions and manage external directory services. As installed, the user ID is <code>Administrator</code> and the login for the user account is disabled.</p> <p>Important: The <code>SysAdmin</code> user must log in and change the password of this account for the user to access My webMethods Server.</p>
System Admin	<p>The system administrator for My webMethods Server. This user can manage My webMethods Server, including analysis, configuration, content, and user management. As installed, the user ID is <code>SysAdmin</code> and the password is specified during installation. This administrator does not use the My webMethods user interface.</p> <p>Important: Depending on the options selected during installation, My webMethods Server might require this user to provide a new password during the first login.</p>
Deleted Items	<p>A user account that is used internally by My webMethods Server to store work done by a user with administrative privileges when that user is deleted from the system. As installed, the user account is <code>DeletedItems</code> and the password is "manage".</p> <p>Important:</p>

User Name	Description
	Change the password for this user.
Designer	<p>The page designer for My webMethods Server. This user has privileges for My webMethods Server similar to those of the system administrator. As installed, the user ID is <code>Designer</code> and the login for the user account is disabled. For more information about functions performed by this account, see chapter <i>Server Page Development</i>.</p> <p>Important: The <code>SysAdmin</code> user must log in and change the password of this account for the user to access My webMethods Server.</p>
Guest	An anonymous user. This user can read pages that allow anonymous access, such as the login page. Otherwise, this user cannot read, modify, or delete content unless permission is explicitly granted by an administrator. As installed, the user ID for this user is <code>Guest</code> .
webMethods System	<p>A user account that is used internally by My webMethods Server to invoke web services. My webMethods Server uses this account for web service authentication from one server to another. As installed, the user account is <code>WEBM_SYSUSER</code> and the login for the user account is disabled.</p> <p>Important: Do not delete this user account. The <code>SysAdmin</code> user must log in and change the password of this account for the user to access My webMethods Server. Changes to the password for this account must be provided to administrators for webMethods applications that use it when communicating with My webMethods Server.</p>
webMethods Cluster	<p>A user account that is used internally by My webMethods Server for authentication among servers in a cluster. As installed, the user account is <code>WEBM_CLUSTERUSER</code> and the password is “manage”.</p> <p>Important: Do not delete this user account. Change the password for this user on one node of the cluster and then restart all nodes for the password to take effect.</p>

For security reasons, the `SysAdmin` must change the default passwords of system users with administrative privileges. For information about how to change passwords for these users, see [“Editing Information for a User” on page 152](#).

External Directory Service

In addition to the system directory service, My webMethods Server can support multiple external directory services, allowing you to manage a much larger and diverse group of users. If your company has one or more directory services, My webMethods Server can connect to those services.

In addition, you can use a database as a directory service, or create custom services to connect a directory provider to an external directory service in My webMethods Server. For more information about such services, see *webMethods CAF and My webMethods Server Java API Reference*.

Note:

During login, conditions in which the role cache or group cache calculation involves user or group searches that take a long time can result in poor performance in My webMethods Server and in LDAP servers to which it is connected. For more information, see [“Configuring Role or Group Cache Lifecycle Calculation” on page 335](#).

Setting Up the Internal System Directory Service

No set up is required for the internal system directory service beyond configuring the My webMethods Server database. Configuring the database was required during the installation of My webMethods Server, as described in *Installing Software AG Products*.

Managing External Directory Services

Configuring an External LDAP, ADSI, or ADAM Directory Service

Use the following procedure to configure My webMethods Server to use an external LDAP, ADSI, or ADAM directory service.

➤ **To configure an external LDAP, ADSI, or ADAM directory service**

1. Navigate to the following page:
 - As My webMethods Administrator: **Navigate > Applications > Administration > My webMethods > Directory Services > Directory Services Administration.**
 - As sysadmin: **Administration > User Management > Directory Services Administration > Directory Services Administration.**
2. On the **Create New Directory Service** tab, select the directory service type from the following options:
 - **LDAP** - Lightweight Directory Access Protocol.
 - **ADSI** - Active Directory Service Interfaces.
 - **ADAM** - Active Directory Application Mode.
3. Click **Next**.
4. Fill in the form to configure the new directory service.

For more information about the available configuration properties, see [“LDAP, ADSI, and ADAM Directory Service Properties” on page 113](#).

- At the bottom of the page, click **Finish**.

Tip:

To test your configuration, perform a query to search for users or groups that are defined in the external directory service. For information about how to perform a query, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).

LDAP, ADSI, and ADAM Directory Service Properties

When configuring an external directory service of type LDAP, ADAM, or ADSI, you can configure the following properties.

In the **General** section:

The following table lists the directory service properties you configure in the General section:

Property	Description
Name	Required. The name to identify the external directory service. My webMethods Server uses this name to display the external directory service in the user interface.
Description	A descriptive comment about the external directory service.
Keywords	One or more keywords to use when searching for external directory services.

In the **Cache** section:

The following table lists the directory service properties you configure in the Cache section:

Property	Description
Cache Capacity	Required. The number of database queries to cache. The default is 1000. My webMethods Server deletes the cache entries when the number of cached queries reaches the specified capacity, starting from the oldest entries.
Cache Timeout	Required. The period of time for which queries remain in the cache unless the cache capacity is exceeded. The default is 1 hour. My webMethods Server deletes cache entries when the cache timeout expires, even if the specified cache capacity is not reached.

My webMethods Server saves all cache in memory and clears all cache entries when restarted.

In the **Connection Information** section:

The following table lists the directory service properties you configure in the Connection Information section:

Property	Description
Service Enabled	Enables or disables the directory service. The default is Yes . This service is enabled.
Connection Error Threshold	Required. The maximum number of connection errors to occur before disabling the service. The default is 10.
Auto Reconnect	Attempt to reconnect to the directory server if the service is disabled after reaching the connection error threshold or if the connection to the server is lost due to a network outage or planned maintenance. Enabled by default.
Auto Reconnect Interval	The period of time (in seconds) to wait between subsequent attempts to reconnect. The default is 6.
Provider URL	Required. The URL for the external directory server using the following syntax: <code>ldap://host_name:port_number</code>
Base DN	Required. The root distinguished name to use when querying the directory server. For example, <code>ou=mywebMethods,o=webmethods.com</code>
User DN	The additional user DN to use when searching and loading users.
Groups DN	The additional group DN to use when searching and loading users.
Use Kerberos	Whether to use Kerberos authentication when connecting to the LDAP service. The default is No . Do not use Kerberos. For more information about using directory services with Kerberos, see “Configure Kerberos Authentication for Directory Services” on page 270.
Use Ticket Cache	Whether to use Kerberos credentials cache while the user session lasts. Available only when the LDAP service is configured to use Kerberos Authentication. The default is No . Do not use ticket cache. For more information about configuring Kerberos ticket cache for directory services, see “Configure Kerberos Authentication for Directory Services” on page 270.

Property	Description
Security Principal	Required when not using Kerberos Ticket Cache. The distinguished name required to log in to the external directory server.
Security Credentials	Required when not using Kerberos Ticket Cache. The password required to log in to the external directory server.
Failover URLs	The URL to another LDAP server that My webMethods Server uses for failover if the primary LDAP server, specified in the Provider URL field, fails. Separate multiple values with spaces.
Search Timeout	<p>Required. The maximum amount of time (in seconds) that an LDAP search query can run before it expires. The default is 0 - the query does not expire.</p> <p>Unless you configure the connection timeout in the custom_wrapper.conf file, My webMethods Server uses the Search Timeout to define the timeout of a connection to an LDAP server. For more information about configuring an LDAP server connection timeout, see “Configuring a Connection Timeout for an LDAP Directory Service” on page 118.</p>
Enable Default Wildcard Searches	<p>Required. Enables or disables the use of wildcard characters in directory searches. The default is Yes. Enable default wildcard searches.</p> <p>Disabling wildcard searches might improve performance for large servers. When using wildcards, servers do not use any internal indexes for search performance.</p>
Enable Group Across Directory Service	<p>Required. Indicates whether to query for group membership across all external directory services, configured in My webMethods Server. When you enable this option, the search queries for group membership across all directory services, which degrades the login performance. The default is No. Group Across Directory Service.</p> <p>For more information, see “Group Membership Across Directory Services” on page 119.</p>
Enable GroupQuickSearch	Required for Active Directory. Indicates whether to determine the group membership of an Active Directory user with one query instead of a recursive search. When you enable this option, the search uses one query, which improves the login performance.. Users must belong either to an Active Directory security group, or a regular group. The default is Disabled .
ActiveDirectory Domain URLs	Applies only to Active Directory. Specify multiple Active Directory sub-domain URLs, separated by spaces.

In the **Advanced Object Filters** section:

The following table lists the directory service properties you configure in the Advanced Object Filters section:

Property	Description
User Object Filter	<p>The LDAP filter that My webMethods Server applies to all queries when searching for users. Use a technical LDAP query that limits the type of objects, exposed in My webMethods Server.</p> <p>Note: It is recommended that you examine the My webMethods Server directory debug logs to ensure that the query is working correctly.</p>
Group Object Filter	<p>The LDAP filter that My webMethods Server applies to all queries when searching for groups. Use a technical LDAP query that limits the type of objects, exposed in My webMethods Server.</p> <p>Note: Examine the My webMethods Server directory debug logs to ensure that the query is working correctly.</p>
Use Nested Groups	<p>Enables or disables searches in nested LDAP groups. The default value is No. Do not use nested groups.</p>
Use the Virtual List View Control	<p>Enables or disables the use of the Virtual List View control to retrieve a subset of objects for an LDAP query. The default value is No. Do not use the VLV control. Applies only when the automatic configuration of LDAP server controls is disabled.</p>
Use Server Side Paging Control	<p>Enables or disables the use of the Server-Side Paging control to page the results of an LDAP query. The default value is No. Do not use the Paging control. Applies only when the automatic configuration of LDAP server controls is disabled.</p>
Use Server Side Sorting Control	<p>Enables or disables the use of the Server-Side Sorting control to sort the results of an LDAP query in a particular order. The default value is No. Do not use the Soting control. Applies only when the automatic configuration of LDAP server controls is disabled.</p>
Automatically Configure Server Side Controls	<p>Enables or disables the automatic configuration of LDAP server controls by My webMethods Server. The default value is Yes. Autoconfigure the controls.</p>

In the **User Attributes** section:

The following table lists the directory service properties you configure in the User Attributes section:

Property	Description
User Object Class	Required. The User Object Class attribute for the external directory service. The default is <code>person</code> .
User ID	Required. The User ID attribute for the external directory service. The default is <code>uid</code> .
First Name	Required. The First Name attribute for the external directory service. The default is <code>sn</code> .
Last Name	Required. The Last Name attribute for the external directory service. The default is <code>givenName</code> .
Full Name	Required. The Full Name attribute for the external directory service. The default is <code>cn</code> .
E-mail Address	Required. The Email Address attribute for the external directory service. The default is <code>mail</code> .
Password	Required. The Password attribute for the external directory service.
User Disabled	The name of an attribute in the external directory service that identifies a user as disabled. The default is <code>true</code> .
User Disabled Value Regex	The regular expression to use when evaluating the User Disabled attribute for the external directory service.
UUID	The name of the attribute to use as a universally unique identification attribute of a user. Specify a string of maximum 128 characters, for example <code>cn</code> or <code>email</code> .

Note:

If you change the value of **UUID** for an existing directory service, you must run the `UserDirectory_UpdateUUID` utility to update the UUID value of directory service users.

For more information, see “ [Configuring Universally Unique Identifier \(UUID\) for Users](#)” on page 120.

In the **Group Attributes** section:

The following table lists the directory service properties you configure in the Group Attributes section:

Property	Description
Group Object Class	Required. The Group Object Class attribute for the external directory service. The default is <code>groupofuniquenames</code> .
Group ID	Required. The Group ID attribute for the external directory service. The default is <code>cn</code> .
Group Name	Required. The Group Name attribute for the external directory service. The default is <code>cn</code> .
Group Members	Required. The Group Members attribute for the external directory service. The default is <code>uniquemember</code> .
Group E-mail	Required. The Group Email attribute for the external directory service. The default is <code>mail</code> .

In the **Connection Pool** section:

The following table lists the directory service properties you configure in the Connection Pool section:

Property	Description
Minimum Connections	The minimum number of connections to the external directory server to keep open at all times. The default is 1.
Maximum Connections	The maximum number of connections to the external directory server to keep open at all times. The default is 20.
Maximum Connection Time	The maximum amount of time to keep a connection to the external directory server open, before recycling the connection. The server resets this value for each LDAP search to ensure that an LDAP connection remains open during the search process. The default is 10 minutes.
Clean Up Interval	The time interval for cleaning up expired LDAP connections. The default is 1 minute.

Note:

In some LDAP implementations, the paging cookie is bound to a specific LDAP connection. Make sure that the value for the Maximum Connections property is large enough to handle concurrent LDAP searches and the value for the Maximum Connection Time property is long enough to ensure that searches can finish within the specified time range.

Configuring a Connection Timeout for an LDAP Directory Service

By default, My webMethods Server uses the value of the Search Timeout property to define the timeout of a connection to an LDAP server. However, the primary purpose of the Search Timeout property is to specify the timeout of an LDAP query.

In some cases you might need to specify a different connection timeout value than the value of Search Timeout. To do that, you can configure the `-Dcom.webmethods.portal.portlet.wm_xt_ldapdirsvc.service.connection.timeout` parameter in the `custom_wrapper.conf` file of My webMethods Server. The default value of the LDAP directory service connection timeout parameter is 10 seconds. If you do not configure the connection timeout parameter and the value of Search Timeout is set to 0, My webMethods Server uses the default connection timeout value of 10 seconds.

For more information about Search Timeout and about configuring an LDAP directory service, see [“Configuring an External LDAP, ADSI, or ADAM Directory Service”](#) on page 112.

➤ To configure a connection timeout for an LDAP directory service

1. Go to `Software AG_directory \profiles\MWS_serverName\configuration` and open the `custom_wrapper.conf` file in a text editor.
2. Add the following parameter:

```
wrapper.java.additional.number=
-Dcom.webmethods.portal.portlet.wm_xt_ldapdirsvc.service.connection.timeout=
time_in_seconds
```

where *number* is a unique sequential number depending on the already existing parameter numbers in the file.

3. Save the file and restart My webMethods Server.

Group Membership Across Directory Services

If you have multiple LDAP, ADSI, or ADAM directory services configured on My webMethods Server, the server can query for group membership across all of the configured directory services. This feature is useful if Users need to be in a branch of the Directory Tree that is distant enough from Groups that it is inefficient to have only one directory service mounted at a root that encompasses both users and groups. Instead, you might configure two directory services. One service points at the root of the User branch while the other points at the root of the Group branch. For example, you might have a directory structure similar to this:

```
o=MyCompany, ou=Americas, ou=US, ou=Groups
o=MyCompany, ou=Americas, ou=US, ou=Users
o=MyCompany, ou=Americas, ou=Mexico, ou=Groups
o=MyCompany, ou=Americas, ou=Mexico, ou=Users
and so forth....
```

My webMethods Server would not perform well with a single directory service pointing to `o=MyCompany`. Instead the administrator might create multiple directory services pointing to `ou=Americas` and other regional OUs. But suppose that Groups can have members from multiple regions, as might be common in large international organizations. In that case, it is possible for the membership of a Group to span multiple directory services.

To make it possible to query for group membership across all configured directory services, set **Enable Group Across Directory Service** for each directory service to **Yes. Enable Group Across Directory Service**.

Note:

Enabling this feature can noticeably degrade login performance.

Enabling Universally Unique Identifier (UUID) Resolution

You can enable Universally Unique Identifier (UUID) resolution in My webMethods Server. You must install the UserDirectory_UpdateUUIDPortlet.war file before enabling UUID resolution. For more information about installing the UserDirectory_UpdateUUIDPortlet.war file, see “[Configuring Universally Unique Identifier \(UUID\) for Users](#)” on page 120.

› To enable UUID resolution in My webMethods Server

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Directory Services**.
 - As system administrator: **Folders > Administrative Folders > Administration Dashboard > User Management > Directory Services Administration**.
2. Click the **Enable UUID Resolution** tab.
3. Select the **Enable UUID Resolution** check box, and then click **Apply**.

Configuring Universally Unique Identifier (UUID) for Users

By default, the external directory service configured with My webMethods Server uses DN to identify users. In this case, if a user is transferred from one organization unit to another, My webMethods Server will not be able to recognize that user because of the change in the user's DN.

For example, if a user with commonName as James (cn=James) is working in the marketing unit (ou=marketing) of ABC organization (o=ABC), and if My webMethods Server is configured to use LDAP, James's DN will be dn: cn=James,ou=marketing,o=ABC. If James is transferred to finance unit, the DN for James will be dn: cn=James,ou=finance,o=ABC. After the DN change, My webMethods Server will not be able to recognize James. Even the roles in which James is a member will not be able to locate James in LDAP, and will throw "unable to get role members-cn=James,ou=marketing,o=ABC" warning.

You can configure My webMethods Server to use UUID for identifying users. My webMethods Server system administrator must configure UUID at a time that is suitable for system maintenance.

Important:

If you have a large user base, the `UserDirectory_UpdateUUID` utility might take a long time to re-configure external user directory with My webMethods Server. To remove invalid user records, run the cleanup utility before you run the `UserDirectory_UpdateUUID` utility.

➤ **To configure My webMethods Server to use UUID for identifying users**

1. Enable UUID resolution. For information about enabling UUID resolution, see [“Enabling Universally Unique Identifier \(UUID\) Resolution”](#) on page 120.
2. Configure the external directory service to set the **UUID** property. For information about configuring an external directory service, see [“Configuring an External LDAP, ADSI, or ADAM Directory Service”](#) on page 112.

Important:

If you change the value of UUID property for an existing directory service, you must run the `UserDirectory_UpdateUUID` utility to update the UUID value of directory service users in My webMethods Server.

3. Make sure the `UserDirectory_UpdateUUIDPortlet.war` file is in `Software AG_directory\MWS\components\extras\userdirectory` folder.
4. Navigate to **Folders > Administrative Folders > Administration Dashboard > Configuration > Install Administration > Extras > userdirectory > UserDirectory_UpdateUUIDPortlet.war**.
5. Click **Install Component**.
6. Navigate to **Folders > System > Portlets > UserDirectory_UpdateUUIDPortlet** and open the `UserDirectory_UpdateUUID` utility portlet in a browser.
7. Click **Update UUID** to update the UUID value in My webMethods Server for all the configured directory service users.

The UUID field value will be null for:

- Invalid users
- System users
- DB users

Important:

It is recommended that you run the `UserDirectory_UpdateUUID` utility only once.

Removing Invalid Users

My webMethods Server cannot recognize users who are deleted from a directory service or moved in a directory service. Such users are regarded as invalid users.

As system administrator, you can use the Cleanup_InvalidUsers utility to remove invalid users from My webMethods Server. In addition to removing invalid users, the Cleanup_InvalidUsers utility removes memberships of invalid users from roles in My webMethods Server.

You should run the Cleanup_InvalidUsers utility before you run the UserDirectory_UpdateUUID utility. For more information about the UserDirectory_UpdateUUID utility, see “[Configuring Universally Unique Identifier \(UUID\) for Users](#)” on page 120.

Important:

If you have a large user base, running the Cleanup_InvalidUsers utility might take a long time. You should run the Cleanup_InvalidUsers utility at a time that is suitable for system maintenance.

> To remove invalid users

1. Take a backup of My webMethods Server database.
2. Navigate to **Folders > Administrative Folders > Administration Dashboard > Configuration > Install Administration > Extras > UserDirectory > UserDirectory_UpdateUUIDPortlet.war**.
3. Click **Install Component**.
4. Navigate to **Folders > System > Portlets > UserDirectory_UpdateUUIDPortlet**, and click **Cleanup_InvalidUsers**.
5. In the **Directory Service Name** field, type the name of a directory service configured with My webMethods Server. You can type the single wildcard character * to specify all directory services configured with my My webMethods Server.
6. Select **Only log invalid user entries** and click **Cleanup**.

All invalid users are listed in the *Software AG_directory \MWS\server\default\logs_full_.log* file. Review the file to ensure that the listed users are not active My webMethods Server users.

7. Select **Log and remove invalid user entries** and click **Cleanup**.
8. Restart My webMethods Server.

Configuring an External Database Directory Service

Use the following procedure to configure My webMethods Server to use an external database directory service.

Note:

To use a database directory service, you must first connect to the database as an external data source. For more information, see “[Managing External Data Sources](#)” on page 129.

> To configure an external database directory service

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Directory Services > Create New Directory Service.**
 - As system administrator: **Administration > User Management > Directory Services Administration > Create New Directory Service.**
2. In **Directory Type** field, click **Database**:
3. Click **Next**.
4. On the **Create Database Directory Service** page, specify:

The following table lists the properties, required to configure an external database directory service:

Section	Property	Description
General	Name	Required. A name to identify the external database directory service. My webMethods Server uses this name when it needs to identify the external database directory service in the user interface
	Description	Optional. A descriptive comment about the external database directory service.
Attributes	User ID	Required. The name of the query field containing the user ID value.
	User DN	Required. The name of the query field containing the distinguished name value for the user.
	User First Name	Required. The name of the query field containing the user first name.
	User Last Name	Required. The name of the query field containing the user last name.
	User Full Name	Optional. The name of the query field containing the user full name. If you do not supply an attribute, the full name is derived from the User First Name and User Last Name attributes.
	User E-mail	Optional. The name of the query field containing the user email address.
	User Disabled	Optional. The name of an attribute in the external directory service that identifies a user as being disabled.

Section	Property	Description
	User Disabled Value Regex	Optional. A regular expression used to evaluate the User Disabled attribute for the external directory service.
	Group ID	Required. The name of the query field containing the group ID value.
	Group DN	Required. The name of the query field containing the distinguished name value for the user.
	Group Name	Required. The name of the query field containing the group name.
	Group E-mail	Optional. The name of the query field containing the group email address.
Configuration	Authentication Handler	Required. The page that handles authentication for the database. By default, My webMethods Server provides a clear-text authentication handler.
Database	Datasource	Required. The database to be used as a data store. For a database to appear in the list, you must first use the DataSource Administration page to connect to the external database.
	Query Lookup User by ID	Required. A SQL query that returns a user record based on the user ID. This query must return all user attributes, as described under Attributes in this table.
	Query Authenticate	Required. A SQL query that returns persisted user credentials for authentication.
	Various queries	Optional. You can define several additional SQL queries, as needed. Sample language is provided for each type of query.
Cache	Cache Enabled	Required. Determines whether My webMethods Server will attempt to save the load on the database by using cached data whenever possible. Select: <ul style="list-style-type: none"> ■ Yes to enable caching ■ No to disable caching
	Cache Capacity	Required. The number of database queries you want to cache. The default is 1000.

Section	Property	Description
	Cache Timeout	Required. The length of time that queries should remain in the cache unless the cache capacity is exceeded. The default is 1 day.

- At the bottom of the page, click **Finish**.

Tip:

To test your configuration to ensure you have correctly configured the external directory service, perform a query to search for users or groups that are defined in the external directory service. For instructions on how to perform a query, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).

Allowing Externally Defined Users to Perform Actions from My webMethods

By configuring an external directory service, you provide My webMethods Server with the information it needs to connect to and retrieve information from the external directory service. After configuring the external directory service, users defined in the external directory service will be able to log into My webMethods *but will not have permission to do anything else*. You need to take the following additional steps to allow users to perform actions from My webMethods:

- Create one or more LDAP query roles or database roles to identify the users who should be granted access to My webMethods. For more information about how to create roles for external directory services, see [“Creating a Role to Define Externally Defined Users You Want to Access My webMethods” on page 125](#).
- Add the roles you have created to the My webMethods Users role. For more information, see [“Adding a Role to the My webMethods Server Users Role” on page 126](#).

Creating a Role to Define Externally Defined Users You Want to Access My webMethods

For each external directory service that you configure, create one or more roles that identify the users or groups of users that you want to allow access to My webMethods Server.

- For an LDAP, ADSI, or ADAM directory service, create an LDAP query role.
- For a database directory service, create a database role.

➤ To create a role to identify users who will be granted access to My webMethods

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System Wide > User Management > Roles > Add Role**.

- As system administrator: **Administration Dashboard > User Management > Manage Roles > Add Role.**
2. In the **Role Name** field, specify the name you want to assign to the new role, for example, "SystemX-My webMethods Users."
3. To specify the type of role you want to create, move it to the **Selected Items** box:
 - **LDAP Query Role Provider**—LDAP, ADSI, or ADAM directory service
 - **Database Role Provider**—database directory service
4. Click **Create Role**.
5. Specify the membership for the role. For more information about how to define the membership:
 - For an LDAP, ADSI, or ADAM directory service, see ["Adding an LDAP Query Role" on page 188.](#)
 - For a database directory service, see ["Adding a Database Role" on page 191.](#)
6. Click **Save**.

Adding a Role to the My webMethods Server Users Role

To grant those users who are identified by the roles you created access to My webMethods, add the roles to the My webMethods Server Users role.

➤ To grant users access to My webMethods

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System Wide > User Management > Roles.**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles.**
2. In the **Keyword** text box, type My webMethods Users, and click **Search**.
3. In the search results panel, click **Edit** for the My webMethods Users role.
4. Click the **Members** tab.
5. Click **Edit Members**.
6. Search for the principals you want to add and for each you want to make a member of the My webMethods Users role, move the role name to the **Selected** panel.

7. After you have selected all roles that identify users you want to assign to the My webMethods Users role, click **Apply**.
8. Click **Save** to make the membership changes to the My webMethods Users role.

Updating the Configuration for a Directory Service

After you initially configure an external directory service or database directory service, you might need to update the values you specified for one or more of the properties. Use the following procedure to update the values of properties associated with a directory service.

➤ To update the configuration for a directory service

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Directory Services > List Directory Services**.
 - As system administrator: **Administration Dashboard > User Management > Directory Services Administration > List New Directory Services**.
2. On the **List Directory Services** panel, do one of the following:
 - Click the name of the directory service configuration you want to modify.
 - Click  or  in My webMethods for the directory service configuration and then click **Properties**.
3. Make your changes to the properties for the directory service.
4. After making your changes, click **Apply**.

Updating the Search Order for Directory Services

Some user actions can result in My webMethods Server querying multiple directory services. Use the following procedure to control the order in which My webMethods Server searches the available directory services.

➤ To update the order in which My webMethods Server searches external directory services

1. Navigate to the following page:
 - As My webMethods Administrator: **Navigate > Applications > Administration > My webMethods > Directory Services > Modify Directory Search Order**.
 - As system administrator: **Administration Dashboard > User Management > Directory Services Administration > Modify Directory Search Order**.

My webMethods Server searches the external directory services in the order they are listed in the **Select** list.

2. To reorder the list, move directory services up or down as required.
3. After you set the search order, click **Apply**.
4. Restart My webMethods Server for the changes to take affect.

Note:

Setting the search order does not affect the order in which My webMethods Server displays directory services in lists throughout the user interface.

Disabling User Accounts

You can prevent users from logging into My webMethods Server based on the value of a specified attribute in an external directory service.

Note:

webMethods products that use Common Directory Services for authentication, such as Integration Server and Optimize are affected by this feature. A user disabled in My webMethods Server is disabled in those other products as well.

➤ **To disable user accounts for an external directory service.**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Directory Services > List Directory Services.**
 - As system administrator: **Administration Dashboard > User Management > Directory Services Administration > List New Directory Services.**
2. On the **List Directory Services** panel, do one of the following:
 - Click the name of the directory service.
 - Click  or  in My webMethods for the directory service and then click **Properties**.
3. Locate the properties needed to disable user accounts.
 - For an LDAP, ADSI, or ADAM directory service, look in the **User Attributes** section.
 - For a database directory service, look in the **Attributes** section.
4. In the **User Disabled** field, type the name of the attribute in the external directory service that will determine the User Disabled status.

The exact value is dependent on the external directory service and the class of users you want to disable.

5. In the **User Disabled Value Regex** field, type a regular expression to match against the value of the **User Disabled** property. When the value matches, the user is disabled
6. After making your changes, click **Apply**.

Deleting a Directory Service Configuration

If you no longer want My webMethods Server to have access to users and groups defined in an external directory service, you can delete the configuration information for that external directory service using the following procedure.

➤ To delete the configuration for an external directory service

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Directory Services > List Directory Services**.
 - As system administrator: **Administration Dashboard > User Management > Directory Services Administration > List Directory Services**.
2. Click  or  in My webMethods for the directory service configuration and then click **Delete**.

Managing External Data Sources

On the Data Source Administration page, you can connect to external data sources, such as databases, and make them available to My webMethods Server. Before you can create and use a database directory service, for example, you must configure the data source.

The Data Source Administration page supports connections to Microsoft SQL Server, Oracle, DB2 Universal, Sybase Adaptive Server, or Informix databases. You can also configure ODBC and custom connections.

Note:

Before you configure a data source for connecting to DB2 Universal, Sybase Adaptive Server, or Informix databases, you must have a corresponding database driver for each respective database application. My webMethods Server distribution does not include database drivers for DB2 Universal, Sybase Adaptive Server, or Informix databases.

Adding a Microsoft SQL Server Data Source

To connect a Microsoft SQL Server data source to My webMethods Server, use the following procedure:

➤ **To add a new data source for Microsoft SQL Server databases**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Data Sources > Add DataSource.**
 - As system administrator: **Administration Dashboard > Configuration > DataSource Administration > Add DataSource.**
2. Type a unique **DataSource Name** to be used by My webMethods Server on the **View DataSources** panel.
3. Type a **Display Name** to be used when you identify a data source to use for a database directory service.
4. Select **MS SQL Server** from the **Server Type** list and click **Next**.
5. Type the SQL Server host name.
6. Type the port number used by the SQL Server. The default port is 1433.
7. Type the database name.
8. Type a valid SQL Server user name and password that, at a minimum, has READ access to the database to which you will connect.
9. Click **Submit**.

Adding an Oracle Data Source

To connect an Oracle data source to My webMethods Server, use the following procedure:

➤ **To add a new data source for Oracle databases**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Data Sources > Add DataSource.**
 - As system administrator: **Administration Dashboard > Configuration > DataSource Administration > Add DataSource.**
2. Type a unique **DataSource Name** to be used by My webMethods Server on the **View DataSources** panel.

3. Type a **Display Name** to be used when you identify a data source to use for a database directory service.
4. Select **Oracle** from the **Server Type** list and click **Next**.
5. Type the Oracle host name.
6. Type the port number on which the Oracle host is running. The default port is 1521.
7. Type the instance name (SID) for the database.
8. Type a valid Oracle database user name and password that, at a minimum, has READ access to the database to which you will connect.
9. Click **Submit**.

Adding a DB2 Universal Data Source

To connect a DB2 Universal data source to My webMethods Server, use the following procedure:

➤ To add a new data source for DB2 Universal databases

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Data Sources > Add DataSource**.
 - As system administrator: **Administration Dashboard > Configuration > DataSource Administration > Add DataSource**.
2. Type a unique **DataSource Name** to be used by My webMethods Server on the **View DataSources** panel.
3. Type a **Display Name** to be used when you identify a data source to use for a database directory service.
4. Select **DB2 Universal** from the **Server Type** list and click **Next**.
5. Type the DB2 host name.
6. Type the port number that DB2 is running on.
7. Type the instance name for the database.
8. Type a valid DB2 user name and password that, at a minimum, has READ access to the database to which you are connecting.

9. Click **Submit**.

Adding a Sybase Adaptive Server Data Source

To connect a Sybase Adaptive Server data source to My webMethods Server, use the following procedure:

➤ To add a new data source for Sybase Adaptive Server databases

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Data Sources > Add DataSource**.
 - As system administrator: **Administration Dashboard > Configuration > DataSource Administration > Add DataSource**.
2. Type a unique **DataSource Name** to be used by My webMethods Server on the **View DataSources** panel.
3. Type a **Display Name** to be used when you identify a data source to use for a database directory service.
4. Select **Sybase Adaptive Server** from the **Server Type** list and click **Next**.
5. Type the Sybase Server host name.
6. Type the port number that Sybase Server is running on.
7. Type the instance name for the database.
8. Type a valid Sybase Server user name and password that, at a minimum, has READ access to the database to which you are connecting.
9. Click **Submit**.

Adding an Informix Data Source

To connect a Informix data source to My webMethods Server, use the following procedure:

➤ To add a new data source for Informix databases

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Data Sources > Add DataSource**.

- As system administrator: **Administration Dashboard > Configuration > DataSource Administration > Add DataSource.**
2. Type a unique **DataSource Name** to be used by My webMethods Server on the **View DataSources** panel.
 3. Type a **Display Name** to be used when you identify a data source to use for a database directory service.
 4. Select **Informix** from the **Server Type** list and click **Next**.
 5. Type the Informix host name.
 6. Type the port number on which the Informix host is running.
 7. Type the database name.
 8. Type the Informix server name.
 9. Type a valid Informix user name and password that, at a minimum, has READ access to the database to which you are connecting.
 10. Click **Submit**.

Adding a Generic ODBC Data Source

Note:

My webMethods Server can use any ODBC connection that is manually configured at the operating system level (such as Windows Server). My webMethods Server uses a standard Java JDBC-ODBC bridge driver to connect to the ODBC data sources on the underlying operating system. Consult your Microsoft vendor documentation for details on how to configure an ODBC data source at the operating system level.

To connect a generic ODBC data source to My webMethods Server, use the following procedure:

➤ To add a new data source for generic ODBC databases

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Data Sources > Add DataSource.**
 - As system administrator: **Administration Dashboard > Configuration > DataSource Administration > Add DataSource.**
2. Type a unique **DataSource Name** to be used by My webMethods Server on the **View DataSources** panel.

3. Type a **Display Name** to be used when you identify a data source to use for a database directory service.
4. Select **Generic ODBC** from the **Server Type** list and click **Next**.
5. Type the ODBC data source name that matches the ODBC data source configured at the operating system level.
6. Type a valid user name and password that, at a minimum, has READ access to the database to which you are connecting.
7. Click **Submit**.

Adding a Custom Data Source

Note:

This option is an advanced data source configuration and requires you to specify a valid JDBC driver class name, connection URL, user name, and password. Consult your vendor documentation to get specific instructions on where to locate the proper database drivers for the database application that you wish to connect to from the server.

To connect a custom data source to My webMethods Server, you need to declare the driver JAR file as a fragment of the `com.webmethods.caf.server` bundle and then use the My webMethods Server user interface to add the datasource.

➤ To add a new data source for a custom database

1. If My webMethods Server is running, stop it.
2. Copy the driver `*.jar` files into the `Software AG_directory \MWS\lib` directory.
3. In the `Software AG_directory \MWS\lib` directory, create a text file with the name `driver-name.bnd`, where `driver-name` is the name of the driver `.jar` file.

For example, if you are using the oracle thin driver, `ojdbc6.jar`, the file name is `ojdbc6.bnd`.

4. In the file, provide instructions for the OSGi bundle conversion.

In the following example, replace the values in *italics* as appropriate. For `Bundle-Version`, it is typical to use the version number of the JAR file, but any unique number is valid. You can, in fact, just use the value in this example:

```
# attach as fragment to the caf.server bundle
Fragment-Host: com.webmethods.caf.server
Bundle-SymbolicName: mws.jar.ojdbc6
Bundle-Version: 0.9.0.v${tstamp}
Include-Resource: ojdbc6.jar
-exportcontents: *
```

```
Bundle-ClassPath: ojdbc6.jar
Import-Package: *;resolution:=optional
```

- At a command line prompt, move to the server's bin directory:

```
Software AG_directory\MWS\bin
```

- Type this command:

```
mws.bat -s serverName update-osgi-profile
```

- Restart your My webMethods Server instance.
- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Data Sources > Add DataSource.**
 - As system administrator: **Administration Dashboard > Configuration > DataSource Administration > Add DataSource.**
- Type a unique **DataSource Name** to be used by My webMethods Server on the **View DataSources** panel.
- Type a **Display Name** to be used when you identify a data source to use for a database directory service.
- Select **Custom JDBC** from the **Server Type** list and click **Next**.
- Type the JDBC Connection class for the custom drivers you want to use for the data source connection.
- Type a valid connection URL.
- Type a valid user name and password that, at a minimum, has READ access to the database to which you are connecting.
- Click **Submit**.

Modifying a Data Source

Note:

You cannot modify the default data source, which is the My webMethods Server database.

To modify a data source connection to My webMethods Server, use the following procedure:

- **To modify an existing data source**

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Data Sources > View DataSources.**
 - As system administrator: **Administration Dashboard > Configuration > DataSource Administration > View DataSources.**
- Click  ( in My webMethods) for the data source you want to modify, and then click **Modify**.
- On the first page of the **Datasource Properties** wizard, specify:

The following table lists the properties to configure when modifying an existing data source:

Field	Description
Datasource Name	Type a different unique data source name to be used by My webMethods Server on the View DataSources panel.
Display Name	Type a different display name for the data source to be used when you identify a data source to use for a database directory service.
Server Type	From the list, select a different database type.

- Click **Next**.
- In the second page of the Datasource Properties wizard, modify one or more of the fields identify the location, names, and passwords for the data source.
- Click **Submit**.

Deleting a Data Source

Note:

You cannot delete the default data source, which is the My webMethods Server database.

To delete a data source connection to My webMethods Server, use the following procedure:

> To delete a data source

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Administration > My webMethods > Data Sources > View DataSources.**
 - As system administrator: **Administration > Configuration > DataSource Administration > View DataSources.**

- Click  or  in My webMethods for the data source you want to modify, and then click **Remove**.

Managing Email Settings

The E-mail Administration page is used to configure the mail server settings used by the server when processing email.

› To configure an email server to send server notifications

- Navigate to one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > E-Mail Servers**
 - As system administrator: **Administration Dashboard > Configuration > E-mail Administration**.
- On the **E-mail Properties** page, specify:

The following table lists the properties, required to configure e-mail notification settings for My webMethods Server

Property	Description
Transport Protocol	Identifies the email protocol to be used. The default and only valid value is <code>smtp</code> .
SMTP Hosts	Identifies the SMTP server. Specify the server's host name. For example: <code>smtp.server.com</code> . If you specify two or more hosts, type one address per line.
SMTP Port	Identifies the port number. Specify the SMTP server's port number. For example, <code>25</code> .
SMTP Username	Optional. Identifies the user name that My webMethods Server is to supply for authentication. If the SMTP server requires authentication, specify the user name to supply to satisfy the authentication challenge.
SMTP Password	Optional. Identifies the password associated with the SMTP Username . If the SMTP server requires authentication, specify the appropriate password.
SMTP TLS Enabled	Optional. Indicates whether to use an encrypted SMTP connection (by means of TLS). If the SMTP server requires authentication, set to one of the following: <ul style="list-style-type: none"> <code>true</code> TLS is enabled.

Property	Description
	<ul style="list-style-type: none"> ■ <code>false</code> (default) TLS is not enabled.
SMTP Timeout	Defines the maximum period of time to wait for a response from the server, specified in milliseconds. Default value is 60000.
SMTP Connection Timeout	Defines the maximum period of time for a given SMTP session, specified in milliseconds. Default value is 60000.
SMTP Debug Enabled	<p>Optional. Indicates whether to enable debugging for email activities. My webMethods Server writes the debugging information to the My webMethods Server logs. Set to one of the following:</p> <ul style="list-style-type: none"> ■ <code>true</code> Debugging is enabled. ■ <code>false</code> (default) Debugging is not enabled.
From Name	<p>Defines the default "From" name. Specify the default name to use in the "From" field of the email messages that My webMethods Server sends using the SMTP server.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: Text in this field is subject to the requirements of the RFC822 Internet Text Message standard. For example, text in parentheses, as in "(Important)", is treated as a comment and is removed when the message is created, and bracketed text, such as "[Status]", is treated as an optional element and is also removed.</p> </div>
From E-Mail Address	Defines the default "From" email address. Specify the default email address to use in the "From" field of the email messages that My webMethods Server sends using the SMTP server.
Skin	Optional. Identifies the skin to use when rendering My webMethods Server email notifications. Specify a My webMethods Server skin.
Admin E-mail Address	Defines the email address of the My webMethods Server administrator. This is used as the 'from' address for administrative email messages sent on behalf of the server.

3. Click **Save Settings**.

My webMethods Server and Multi-Factor Authentication

You can configure My webMethods Server to require two-factor authentication with a time-based one-time password (TOTP). My webMethods users can get a temporary pass code through a preconfigured authenticator application, and supply the code on a dedicated screen, before logging in to My webMethods with their credentials.

Alternatively, you can develop a custom authentication scheme and customize the login page for My webMethods to require both the TOTP code and user credentials on a single screen using the My webMethods Server API. For more information, see *webMethods CAF and My webMethods Server Java API Reference*.

Implementing a two-factor authentication flow with a time-based one-time password in My webMethods Server requires the following:

- A valid e-mail server configuration. For information about how to configure an e-mail server for My webMethods Server, see [“Managing Email Settings” on page 137](#).
- A valid e-mail address, configured on the **User Information** page for every user that must access My webMethods using a time-based one-time password.
- A valid TOTP configuration.
- For each My webMethods user, an authenticator application, configured with the shared secret that My webMethods Server generates on the **User Information** page for the user.

Configuring Multi-Factor Authentication with Time-Based One-Time Passwords

Before you configure My webMethods Server to require two-factor authentication with a time-based one-time password (TOTP), you must configure an e-mail server. Users that access My webMethods with their credentials and a temporary password must have valid e-mail addresses, configured on their **User Information** page, and an authenticator application, configured for TOTP in My webMethods Server. Although you can register My webMethods Server with different one-time password services or authenticator applications, you can only have one active TOTP configuration at a time.

➤ To configure My webMethods Server for TOTP Authentication

1. Navigate to **Applications > Administration > My webMethods > One-time passwords Administration**.
2. On the **One-time password configuration properties** screen, configure the following:
 - **Name** - Required. A unique name for the TOTP configuration.
 - **Service enabled** - The default value is **No. This service is disabled**. Select **Yes. This service is enabled**. from the drop-down list to enable the TOTP service and require one-time password from My webMethods users at login.
 - **Time-step windows** - The number of rolling windows to accept when validating the code. Determines the overall validity time for the one-time pass code. The default value is **3** (three time-step windows).
 - **One-time password role name** - The name of the role to use when registering users for TOTP authentication. The default value is **TOTPSinkRole**. My webMethods Server

automatically adds users to this role when they generate their shared secret from the **User Information** page for their profile.

- **One-time password service name** - The display name of the one-time password service provider or authenticator application.
- Save the configuration.

After you configure TOTP, each My webMethods user must generate a shared secret from the **User Information** page for the user profile, and register with the one-time password service or authenticator application. For more information about generating a shared secret, see *Working with My webMethods*.

Managing Calendars

Software AG Designer and My webMethods Server support the use of business calendars and user calendars to assist with task definition, assignment, and behavior. Both business and user calendars are set up and configured in My webMethods Server. Each type of calendar is configured separately, and you can define business calendars only, user calendars only, or both.

User and business calendars are most often used by the Task Engine in the assignment and scheduling of tasks. For complete information about creating and managing user and business calendars, see *webMethods Task Engine User's Guide*.

9 Searches for Users, Groups, and Roles

■ Searching for Existing Users, Groups, or Roles	142
■ Advanced Searches	142
■ Working with Saved Searches	144
■ Exporting Search Results to a .csv File	146

Searching for Existing Users, Groups, or Roles

You can search for the following:

- Users and groups defined in the internal system directory service
- Users and groups defined in external directory services
- Roles

➤ **To search for users, groups, or roles**

1. To navigate to the correct page, do one of the following:

- In My webMethods: **Navigate > Applications > Administration > User Management > *User_type* > Search.**
- As system administrator: **Administration Dashboard > User Management > Manage *User_type* > Search.**

where ***User_type*** is **Users, Groups, or Roles.**

2. For users and groups, from the **Name** list select the directory service where the users or groups are defined.

If you select **Any Directory**, all directory services connected to the server are searched.

3. In the **Keyword** field of the **Search** panel, specify the search criteria, as follows:

- To search for users, enter the first name, last name, e-mail address, or user ID.
- To search for groups, enter the group name, group ID, or group e-mail address.
- To search for roles, enter the role name or the role ID.

The search is *not* case sensitive. If you leave the text box blank, My webMethods Server returns information for all entries in the selected directory service.

4. Click **Go**.

My webMethods Server displays the results in a table format.

For information on exporting the results of a search, see [“Exporting Search Results to a .csv File” on page 146.](#)

Advanced Searches

As a system administrator, you can perform an advanced search for users or groups, based on user or group information and extended attributes. You cannot perform an advanced search for roles.

➤ **To perform an advanced search for users or groups**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > User Management > User_type > Search.**
 - As system administrator: **Administration Dashboard > User Management > Manage User_type > Search.**

where **User_type** is **Users, Groups,** or **Roles.**

2. For users or groups, from the **Directory Service** list select the directory service where the users or groups are defined.

If you select **Any Directory**, all directory services connected to the server are searched.

3. For users, in the **Core Attributes** panel, fill in any of the following fields that are part of User Information:

The following table lists the core attributes for users:

Field	Description
First Name	Type a first name for the saved search.
Last Name	Type a last name for the saved search.
User ID	Type a user ID for the saved search.
E-mail Address	Type a user email address for the saved search.

4. For groups, in the **Core Attributes** panel fill in any of the following fields that are part of Group Information:

The following table lists the core attributes for groups:

Field	Description
Group Name	Type a group name for the saved search.
Group ID	Type a group ID for the saved search.
E-mail Address	Type a group email address for the saved search.

5. In the **Extended Attributes** panel, choose an extended attribute by doing the following:
 - a. From the **Attribute Provider** list, choose an attribute provider.

For more information on attribute providers, see [“Attribute Providers” on page 241](#).

- b. In the **Attribute Value** field, type or modify the value to be used in the saved search.
6. Click **Go**.

My webMethods Server displays the results in a table format.

For information on exporting the results of a search, see [“Exporting Search Results to a .csv File” on page 146](#).

Working with Saved Searches

If you perform a particular search regularly, you can save search criteria that you can reuse.

Creating a Saved Search

To create a saved search for a user, group, or role, follow these steps:

> To create a saved search

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > User Management > User_type > Search**.
 - As system administrator: **Administration Dashboard > User Management > Manage User_type > Search**.

where **User_type** is **Users, Groups, or Roles**.

2. For users and groups, in the **Directory Service** list choose the directory service that contains the users you want to find.

If you select **Any Directory**, all directory services connected to the server are searched.

3. In the search field of the **Search** panel, specify one of the following:
 - All or part of the user ID - to search for users.
 - All or part of the group ID - to search for groups.
 - All or part of the role name - to search for roles.
4. Click **Save**.
5. In the **Search Name** field of the Save Searches dialog box, type a name by which you can identify the search criteria and click **OK**.

Using a Saved Search

You can use a saved search to find users, groups, or roles that match the criteria.

> To perform a saved search

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > User Management > User_type > Saved.**
 - As system administrator: **Administration Dashboard > User Management > Manage User_type > Saved.**
where **User_type** is **Users, Groups, or Roles.**
2. In the **Saved Search** list, choose the name of the saved search and click **Go.**

Modifying a Saved Search

You can modify an existing saved search.

> To modify a saved search

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > User Management > User_type > Saved.**
 - As system administrator: **Administration Dashboard > User Management > Manage User_type > Saved.**
where **User_type** is **Users, Groups, or Roles.**
2. In the **Saved Search** list, choose the name of the saved search to be modified and click **Details.**
3. Do any of the following:
 - In the search field of the **Saved** panel, change the search criteria.
 - For users and groups, in the **Directory Service** list change the directory service in which to perform the search.
 - For users, in the **Core Attributes** panel add, modify, or remove any of the following fields:
 - **First Name**
 - **Last Name**

- **User ID**
- **E-mail Address**

The core user attributes match the user attribute fields on the **User Information** panel, described in [“User Information” on page 153](#).

- For groups, in the **Core Attributes** panel add, modify, or remove any of the following fields:

- **Group Name**
- **Group ID**
- **E-mail Address**

The core user attributes match the group attribute fields on the **Group Information** panel described in [“Group Information” on page 164](#).

- In the **Extended Attributes** panel, choose an extended attribute by doing the following:
 1. From the **Attribute Provider** list, choose an attribute provider.
For more information on attribute providers, see [“Attribute Providers” on page 241](#).
 2. In the **Attribute Value** field, type or modify the value to be used in the saved search.

4. To update the saved search, click **Save**.

Deleting a Saved Search

When you no longer need a saved search, you can delete it.

> To delete a saved search

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > User Management > User_type > Saved**.
 - As system administrator: **Administration Dashboard > User Management > Manage User_type > Saved**.where *User_type* is **Users, Groups, or Roles**.

2. In the **Saved Search** list, choose the name of the saved search to be deleted and click **Delete**.

Exporting Search Results to a .csv File

You can export search results to a comma-delimited text file (.csv file) if the search results panel includes the **Export Table** function.

After exporting search results to a .csv file, you can then import the .csv file into Microsoft Excel, Microsoft Access, or any other application that accepts the .csv file format.

➤ **To export search results**

1. In the search results panel, click **Export Table**.
2. From the **Character Encoding** list, select the character encoding to use.
3. Click **Export**.
4. Use the file-download mechanism in your browser to browse to the location where you want to save the .csv file.

10 Managing Users and Groups

■ About Managing Users and Groups	150
■ Managing Users	151
■ Managing User Data	161
■ Managing Groups	163

About Managing Users and Groups

You can manage users and groups as My webMethods Administrator or as a system administrator (or as a user with administrator privileges, as described in [“About Roles in My webMethods Server”](#) on page 184).

Users

To access My webMethods, the system must have access to a definition for the user. To define users, you can:

- **Add users to the internal system directory service.** You provide all information about the user, for example the user ID the user is to supply to log into the system and the user’s password. For information about how to manage users in the internal system directory, see [“Managing Users”](#) on page 151.
- **Access users already defined in external directory services.** If your users are defined in one or more external directory services, you can configure My webMethods Server to use the external directory services. As a result, those users can access and use My webMethods. For more information, see [“Managing External Directory Services”](#) on page 112.

Note:

With My webMethods Server, you can use a combination of users that are defined in both the internal system directory service and external directory services.

Groups

You can logically organize collections of users into groups, which allows you to identify a group of users by a group name rather than identifying each user individually. For example, if you want to assign a group of users to a role, you can simply assign the group containing the users to the role, rather than identifying each user individually.

To define a group, you can do the following:

- **Add groups to the internal system directory service.** You provide information about the group and define its membership. You can assign both individual users or other groups to be members. The users and groups that you assign to a group that is defined in the internal system directory service must also be defined in the internal system directory service. That is, you cannot assign users or groups that are defined in an external directory service to an internally-defined group. For information about how to manage groups in the internal system directory, see [“Managing Groups”](#) on page 163.
- **Access groups already defined in external directory services.** If you want to use groups that are defined in one or more external directory services, you can configure My webMethods Server to use the external directory services. For more information, see [“Managing External Directory Services”](#) on page 112.

Note:

With My webMethods Server, you can use a combination of groups that are defined in both the internal system directory service and external directory services.

Managing Users

Adding Users

Before a user can access and use My webMethods, you must either add the user to the internal system directory service or configure My webMethods Server to use the external directory service where the user is defined. For more information about configuring an external directory service, see [“Managing External Directory Services” on page 112](#). The following procedure describes how to add a user to the internal system directory service.

Note:

When a My webMethods Administrator adds a new system user, that user is automatically added to the My webMethods Users role. When a system administrator adds a new user, that user is not automatically added to the My webMethods Users role.

> To add a user

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Users > Add User.**
 - As system administrator: **Administration Dashboard > User Management > Manage Users > Add User.**
- Fill in the following fields for the user you want to add to the internal system directory service:

The following table lists the required and optional fields to configure when adding a new user to the system directory service:

Field	Description
User ID	<p>The user ID that you want to assign to the user you are adding. My webMethods Server uses the user ID when forming the distinguished name (DN) for the user.</p> <p>The user ID can be 1 through 226 characters and can contain only alphanumeric ASCII characters with no spaces. The user ID is not case sensitive.</p> <p>Note: My webMethods Server adds the user ID to the internal system directory service using the case you specify. My webMethods Server typically regards user IDs as case-insensitive; however,</p>

Field	Description
	My webMethods Server uses the case you specify for actions that are case-sensitive, for example, HTTP authentication.
Password	The password for the new user.
Confirm Password	The same password you specified in the Password field.
First Name	The first name of the user you are adding. My webMethods Server uses the user's first and last name when displaying the user's name on pages in the user interface.
Last Name	The last name of the user you are adding.
E-mail Address	(Optional) The email address for the user you are adding. My webMethods Server uses the email address when it needs to send a notification to the user by means of an email message.

3. Click **Create**.

Editing Information for a User

You can edit the information for a user defined in the internal system directory service. If a user is defined in an external directory service, you can edit only My webMethods Server-specific information. You must update the external directory service directory to change settings that My webMethods Server obtains from the external directory. For a list of the fields that My webMethods Server maintains for a user, see [“User Information” on page 153](#).

➤ To edit a user

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Users**.
 - As system administrator: **Administration Dashboard > User Management > Manage Users**.
2. Search for the user you want to edit. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).
3. In the search results, click any link in the row of the user you want to edit or click .
4. Make the changes you want to the user's information and click **Save**.

For a description of all the fields on each panel, including whether you can update a field or whether a field is view-only, see [“User Information” on page 153](#).

Important:

My webMethods Server displays the information grouped on various panels. After making changes to information on a single panel, be sure to click **Save** to save your changes *before* selecting another panel.

User Information

Both My webMethods Administrator and sysadmin users can edit the information for a user. User attributes on the **Edit User** page in My webMethods differ from sysadmin interfaces.

The following tables list the information that My webMethods Server maintains for a user.

Note:

The **Default** setting for all locale attributes represents a null value.

User Information

The **User Information** section includes all attributes that you specify when adding a user to the internal system directory service. You cannot edit the **User Information** attributes for users in an external directory service. This section is part of the user's profile, and users can update some of the fields, for example the password and email address. The following table lists the attributes, that the **User Information** section includes:

Field	Description
User ID	Required when adding a new user. The ID that a user supplies to log in to My webMethods. You cannot update this field.
Password	Required when adding a new user. The password that a user must supply to log in to My webMethods. My webMethods Server masks this field in the user interface. To change the password for a user, update the Password and Confirm Password fields.
Confirm Password	Required when adding a new user. The same as the Password field.
First Name	Required when adding a new user. The first name of the user, without the following special characters: >, ', ", and &.
Last Name	Required when adding a new user. The last name of the user, without the following special characters: <, >, ', ", and &.
E-mail Address	The email address that My webMethods Server uses to send notifications to the user, without the following special characters: <, >, ', ", and &.
Distinguished Name (DN)	The distinguished name for the user. My webMethods Server generates this value using the user information you provide. You cannot update this field.

Field	Description
Login Disabled	Applies only to users in the internal system directory. Indicates whether the user is allowed to log in to My webMethods.

User Attributes

The **User Profile** tab includes attributes that My webMethods Server maintains regardless of the directory service to which the user belongs. The following table lists the attributes that the **User Profile** tab includes:

Field	Description
First Name	The first name of the user from the User Information section. You cannot edit this field on the User Profile tab. To change the first name, edit the value in the User Information section.
Last Name	The last name of the user from the User Information section. You cannot edit this field on the User Profile tab. To change the last name, edit the value in the User Information section.
Middle Name	The middle name of the user.
Title	The title of the user, for example, Mr., Mrs., or Ms. If the title you want to use is not available in the list, select Other , and enter a custom value.
Name Suffix	The suffix that appears after the name of the user, if applicable, for example, Jr., Sr., PhD, III. If the suffix you want to use is not available in the list, select Other and enter a custom value.
Preferred Language/Locale	The language and locale for the user.
Address 1 Address 2	The street address for the user.
Custom Address	Additional information included when more than a postal code is required for the address of the user, for example, special instructions.
City	The city where the user is located.
State/Province	The state or province where the user is located.
Postal Code	The postal code for the user, for example, a ZIP Code if the user is located in the United States.
Country/Region ID	The country where the user is located.
Phone 1 Area Code	The area code for the user.
Phone 1 Number	The phone number for the user.

Field	Description
Phone 1 Extension	The extension at which the user can be reached, if applicable.
Phone 1 Country Code	The country code associated with the phone number of the user.
Default Date Format	The date format that My webMethods uses when displaying dates. For more information about date formats, see the Javadoc for SimpleDateFormat.
Default Time Format	The time format that My webMethods uses when displaying time. For more information about time formats, see the Javadoc for SimpleDateFormat.
Default Time Zone	The time zone that My webMethods uses when displaying times.
Default Number Format	The format that My webMethods uses when displaying numbers. For more information about number formats, see the Javadoc for DecimalFormat.
Default Currency Format	The format that My webMethods uses when displaying currency. For more information about currency formats, see the Javadoc for DecimalFormat.
Android Subscription	Displays Android device subscribed to receive mobile notifications for the user profile.
iOS Subscription	Displays the iOS device subscribed to receive mobile notifications for the user profile.
Created Date	Displays the date when the user profile was created. You cannot update this field.
Last Modified Date	Displays the date when the user profile was last modified from the Edit User page. You cannot update this field.

The **User Preferences** tab lists preferences for the display of information in My webMethods. Users can update preferences from their profiles. For more information, see *Working with My webMethods*. The following table lists the attributes, that the **User Preferences** tab includes.

Field	Description
Start Page	<p>Indicates the start page for a profile. To modify, select one of the following options:</p> <ul style="list-style-type: none"> ■ Application Page - an application page to open as a start page. ■ Workspace - a workspace to open as a start page. ■ Clear - restore the default settings.

Field	Description
Items Per Page	The number of items to include on one page when displaying items in a table for this user.
Close Open Tabs on Login	Indicates whether My webMethods Server closes all open tabs from the previous session when the user logs on again.
Open Last Active Tab on Login	Indicates whether My webMethods Server opens the last active tab from the previous session when the user logs on again.
Show Delete Workspace Confirmation	Indicates whether My webMethods Server displays a confirmation dialog box when the user deletes a workspace.
Show Delete Window Confirmation	Indicates whether My webMethods Server displays a confirmation dialog box when the user deletes a window from a workspace.
Show Close All Tabs Confirmation	Indicates whether My webMethods Server displays a confirmation dialog box when the user closes all open tabs.
Auto-Save when Navigating to a Different Workspace	Indicates whether My webMethods Server automatically saves the current workspace when the user navigates to another workspace.

The **Roles** tab displays the roles to which a user is assigned and the dynamic attributes associated with each role. For more information about roles and dynamic role attributes, see [“Creating Roles” on page 186](#) and [“Defining Dynamic Attributes Associated with a Role” on page 194](#).

Note:

If a user is assigned to dynamic roles, the list of roles might not always be completely accurate. My webMethods Server evaluates attributes and determines the roles to which a user belongs when a user logs in to My webMethods. If a change in user attributes occurs during a user session, and the user is no longer eligible to be a member of a role, My webMethods Server continues to consider the user a member of the role until the next login.

The following table lists the attributes, that the **Roles** tab includes:

Field	Description
Role Precedence	The roles to which the user is assigned, listed in order of precedence. Move roles up or down to reorder.
Role Member Attributes	The dynamic attributes associated with the selected role. Dynamic attributes provide more information about a role. For each attribute, the following information is available: <ul style="list-style-type: none"> ■ Attribute - the display name that you specify when adding the attribute to the role. ■ Data Type - the data type of the attribute.

Field	Description
	<ul style="list-style-type: none"> ■ Role Value - the default value for the attribute. All users are assigned this value unless a user-specific value takes precedence. ■ User Value - the user-specific value for the attribute. <p>For example, for a “Customer Service” role, you can add a “Location” attribute to identify where the user assigned to the “Customer Service” role is located. For more information about modifying role and user attributes, see “Defining Dynamic Attributes Associated with a Role” on page 194 and “Setting User-Specific Values for Dynamic Attributes” on page 195.</p>

The **Calendar** tab displays information about the user calendars for business or personal use. For information about creating and managing user and business calendars, see *webMethods Task Engine User’s Guide*. The following table lists the attributes that the **Calendar** tab includes:

Field	Description
User Business Calendar	The calendar to use as a business calendar.
User Personal Calendar	The calendar to use as a personal calendar.

The **LDAP Attributes** tab - If a user is defined in an external directory service, this tab lists a set of specific attributes from the external directory service. LDAP attributes must be set by the sysadmin. For more information, see [“Exposing LDAP Attributes from an External Directory Service” on page 245](#).

Database Attributes tab - If the user is defined in an external database directory service, this tab lists a set of specific attributes from the external database directory service. Database attributes must be set by the sysadmin. For more information, see [“Exposing Database Attributes from an External Directory Service” on page 246](#).

The **Groups** tab displays the groups to which the user belongs, and allows modification of the group membership for the user. The following table lists the attributes, that the **Groups** tab includes:

Field	Description
Location	Indicates the directory from which My webMethods Server obtains the information about the group membership of the user.
Group Membership	The group membership options, available for selection. Use the arrow buttons to move an option to the Selected Items field, and click Apply to add the new group.
Selected Items	The groups to which the user is assigned, if available. Use the arrow buttons to move an option to the Group Membership field, and click Apply to remove group membership for the user.

Exporting User Data

You can export the profile data using the My webMethods user interface or an API call.

My webMethods Server exports personal data for a user in .json file format.

➤ To export user data in My webMethods

1. Navigate to **Applications > Administration > System-Wide > User Management > Users**.
2. In the **Users** list, click the user name for the user for which you want to export profile data.
3. On the **User Profile** tab of the **User Attributes** section, click **Export User Data**.

Exporting User Data with the My webMethods Server REST API

To export the data for a user using the My webMethods Server REST API, make a GET request to the following endpoint:

```
http://server_url:8585/api/users/personalData/userID
```

Usage Notes

This API requires administrator privileges.

Examples

The URL to use to export the profile data of a user with ID `myUser`, from a My webMethods Server running on port 8585 of the host `mws.company.com`:

```
http://mws.company.com:8585/api/users/personalData/myUser
```

Assigning a User to a Group

You can assign users that you have defined in the internal system directory service to groups that you have also defined in the internal system directory service. For information on creating groups, see [“Adding Groups” on page 163](#).

Note:

You cannot assign users that are defined in an external directory service to a group defined in the internal system directory. Similarly, you cannot assign users defined in the internal system directory service to an externally-defined group. You can, however, assign both internal and external users to a role. See [“Managing Roles and Access to My webMethods” on page 183](#).

➤ To assign a user in the system directory service to a group in the system directory service

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Users.**
 - As system administrator: **Administration Dashboard > User Management > Manage Users.**
2. Search for the user that you want to assign to a group. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#). Be sure to select **system** from the **Name** list.
3. In the search results, click any link in the row of the user you want to edit or click .
4. Click **Groups**.
5. For each group to which the user should be a member, move it to the **Selected Items** box.
6. With all groups in the **Selected Items** box, click **Save (Apply** in system administration).

Disabling Login for a User

You can deny a user defined in the internal system directory service the ability to log into My webMethods Server. To disable login for users defined in an external directory service, see [“Disabling User Accounts” on page 128](#).

Note:

webMethods products that use Common Directory Services for authentication, such as Integration Server and Optimize are affected by this feature. A user disabled in My webMethods Server is disabled in those other products as well.

> To disable log-in for a user

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Users.**
 - As system administrator: **Administration Dashboard > User Management > Manage Users.**
2. Search for the user you want to edit. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).
3. In the search results, click any link in the row of the user you want to disable or click .
4. Select the **Login Disabled** option and click **Save**.

Deleting a User

You can remove users that you have previously defined in the internal system directory service.

➤ To delete users from the internal system directory service

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Users.**
 - As system administrator: **Administration Dashboard > User Management > Manage Users.**
2. Search for the users that you want to delete. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#). Be sure to select **system** from the **Name** list.
3. In the search results, select the check boxes beside the user IDs for the users you want to delete, and click **Delete**.

Deleting a User with the My webMethods Server REST API

To delete a My webMethods user using the My webMethods Server REST API, make a DELETE request to the following endpoint:

```
http://server_url:8585/api/users/userID
```

Usage Notes

- This API requires administrator privileges.
- The LDAP and database user you delete with this API will be able to log in to My webMethods until removed from the external directory.

Examples

The URL to use when deleting the profile of a user with ID `myUser`, from a My webMethods Server running on port 8585 of the host `mws.company.com`:

```
http://mws.company.com:8585/api/users/myUser
```

Locating a User’s Home Folder (System Administrator Only)

A system administrator can locate and browse to a user's personal folders. Use this feature if items become unavailable to the user because of permissions changes, or to remove content when a user is no longer actively using the server.

➤ **To locate a user's Home folder**

1. As system administrator: **Administration Dashboard > User Management > Locate a User's Home Folder.**
2. Click **Browse.**
3. To select a user, move that user to the **Selected Items** box and click **Select.**
4. To open the home folder for the user you selected, click **Apply.**

Managing User Data

Data protection laws and regulations, such as the GDPR (General Data Protection Regulation) might require specific handling of user data, even after a user profile is removed.

To ensure user data integrity, My webMethods Server stores timestamps for the initial registration of a of a user profile in My webMethods, and the latest modification that occurred before the current login session. Both timestamps are available for users and administrators to view as part of the user information for the profile. My webMethods users can review profile timestamps to ensure that they are aware of all profile modifications.

When an administrator deletes a profile using the user interface or API, or when users delete their own profiles, My webMethods Server clears all information about the user from the database and cache. However, since deleting a user account does not remove user identifying data that appear in log files, administrators can use the timestamps as a reference when cleaning up My webMethods Server log data.

Additionally, administrators can configure My webMethods Server to send e-mail notifications to users when their profile was modified, either by the users or by an administrator, and determine the level of detail that notification e-mails include.

For more information about GDPR and data protection configurations, see:

- [“Configuring GDPR Settings” on page 161](#)
- [“User Information” on page 153](#)
- [“Exporting User Data” on page 158](#)
- [“Deleting a User” on page 160](#)
- [“Controlling Server Logging” on page 306](#)

Configuring GDPR Settings

You enable and disable GDPR-compliant user data protection options from the **GDPR Configuration** page in My webMethods.

1. As Administrator, navigate to **Applications > Administration > System-Wide > User Management > GDPR Configuration**.
2. Check **GDPR enabled** to enable GDPR options.
3. Optionally, configure e-mail notifications by performing the following steps:
 - a. From the **Select Notification Type** drop-down list, select the user profile event, for which you want to configure e-mail notifications.

Available notification types are **Update Notifications** and **Delete Notifications**

- b. From the **Notification Enabled** drop-down list, enable or disable e-mail notifications for the event.
- c. In the **Notification Subject** field, enter a custom subject, or leave the default text.
- d. In the Notification Content field, modify the message body as required.

Strings enclosed in double braces are variables that My webMethods Server uses to include the user names of My webMethods users in the message. The `{{user}}` variable denotes the user that receives the message, the `{{actor}}` variable - the user that makes modifications to the profile, and the `{{changes}}` variable - details about the profile modifications, if configured. You can move the variables through the text, as in the following example:

Default message:

```
Dear {{user}},
Your personal data has been modified.
{{changes}}
The account was updated by user logged-in as {{actor}}.
```

Modified message:

```
Hello {{user}},
You are receiving this message because
{{actor}} has modified your profile data.
Modifications:
{{changes}}
```

4. Optionally, configure what information to include about the modified profile attributes. Details replace the `{{changes}}` variable in the notification message. The options are:
 - **No details** - Default. Do not include detailed information about modified attributes.
 - **Include changed attribute names** - Include only the names of modified attributes in the notification message.
 - **Include changed attribute with values** - Include the names and values of modified attributes in the notification message.

5. Click **Save**.

You configure notification settings and message texts separately for each notification type.

Managing Groups

Adding Groups

You can define groups of users in the internal system directory service. To do so, first create the group. After the group is created, you can add members to the group.

> To create a group

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Groups > Add Group**.
 - As system administrator: **Administration Dashboard > User Management > Manage Groups > Add Group**.
2. Fill in the following fields for the group you want to add to the internal system directory service:

The following table lists the required and optional fields to configure when adding a new group to the system directory service:

Field	Description
Group ID	An ID for the group. My webMethods Server uses this ID in the distinguished name (DN) for the group. The group ID can be 1 through 255 characters and can contain only alphanumeric ASCII characters with no spaces. The group ID is not case sensitive.
Group Name	The name that you want to assign to the group you are adding. The group name can be 1 through 255 characters.
E-mail Address	(Optional) The email address for the group you are adding.

3. Click **Create**.

Editing Group Information

You can edit the information for a group defined in the internal system directory service. If a group is defined in an external directory service, you must update the external directory service directory

to change settings that My webMethods Server obtains from the external directory. For a list of the fields that My webMethods Server maintains for a group, see [“Group Information” on page 164](#).

> To edit a group

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Groups**.
 - As system administrator: **Administration Dashboard > User Management > Manage Groups**.
- Search for the group you want to edit. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).
- In the search results, click any link in the row of the group you want to edit or click .
- Make the changes you want to the group information and click **Save (Apply** in system administration).

For a description of all the fields on each panel, including whether you can update a field or whether a field is view-only, see [“Group Information” on page 164](#).

Important:

My webMethods Server displays the information grouped on various panels. After making changes to information on a single panel, be sure to click **Save (Apply** in system administration) to save your changes *before* selecting another panel.

Group Information

The following table lists the information that My webMethods Server maintains for a group. The Panel column of the table lists the panel on the **Edit Group** page where the field is maintained.

Panel	Description								
Group Information	Attributes that you specify when you add a group.								
	<table border="1"> <thead> <tr> <th>Fields</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Group ID</td> <td>The group ID assigned to a group. The group ID is defined when the group is added and cannot be changed.</td> </tr> <tr> <td>Group Name</td> <td>The group name for the group.</td> </tr> <tr> <td>E-mail Address</td> <td>The email address for the group.</td> </tr> </tbody> </table>	Fields	Description	Group ID	The group ID assigned to a group. The group ID is defined when the group is added and cannot be changed.	Group Name	The group name for the group.	E-mail Address	The email address for the group.
Fields	Description								
Group ID	The group ID assigned to a group. The group ID is defined when the group is added and cannot be changed.								
Group Name	The group name for the group.								
E-mail Address	The email address for the group.								

Panel	Description										
	<p>Distinguished Name (DN) The distinguished name for the group. You cannot update this field. My webMethods Server forms this field using information defined for the group.</p>										
Groups	Groups to which the current group belongs.										
	<table border="1"> <thead> <tr> <th>Fields</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Location</td> <td>Identifies where My webMethods Server obtains the items it displays in the left panel of Group Membership field.</td> </tr> <tr> <td>Group Membership</td> <td>The left panel lists the items available to select.</td> </tr> <tr> <td>Selected Items</td> <td>The right panel, Selected Items, lists the groups to which the group is assigned, if any.</td> </tr> <tr> <td>Search</td> <td>You can use the Search field to help you search for and quickly locate an item in the Group Membership left panel. This is useful when the left panel contains more items than will display on a single page.</td> </tr> </tbody> </table>	Fields	Description	Location	Identifies where My webMethods Server obtains the items it displays in the left panel of Group Membership field.	Group Membership	The left panel lists the items available to select.	Selected Items	The right panel, Selected Items , lists the groups to which the group is assigned, if any.	Search	You can use the Search field to help you search for and quickly locate an item in the Group Membership left panel. This is useful when the left panel contains more items than will display on a single page.
Fields	Description										
Location	Identifies where My webMethods Server obtains the items it displays in the left panel of Group Membership field.										
Group Membership	The left panel lists the items available to select.										
Selected Items	The right panel, Selected Items , lists the groups to which the group is assigned, if any.										
Search	You can use the Search field to help you search for and quickly locate an item in the Group Membership left panel. This is useful when the left panel contains more items than will display on a single page.										
Group Members	The members that are assigned to this group. The members can be users and other groups.										
	<table border="1"> <thead> <tr> <th>Fields</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Location</td> <td>Identifies where My webMethods Server obtains the items it displays in the left panel of Group Members field.</td> </tr> <tr> <td>Group Members</td> <td>The left panel lists the items available to select.</td> </tr> <tr> <td>Selected Items</td> <td>The right panel, Selected Items, lists the users and groups that are members of the group.</td> </tr> <tr> <td>Search</td> <td>You can use the Search field to help you search for and quickly locate an item in the Group Members left panel. This is useful when the left panel contains more items than will display on a single page.</td> </tr> </tbody> </table>	Fields	Description	Location	Identifies where My webMethods Server obtains the items it displays in the left panel of Group Members field.	Group Members	The left panel lists the items available to select.	Selected Items	The right panel, Selected Items , lists the users and groups that are members of the group.	Search	You can use the Search field to help you search for and quickly locate an item in the Group Members left panel. This is useful when the left panel contains more items than will display on a single page.
Fields	Description										
Location	Identifies where My webMethods Server obtains the items it displays in the left panel of Group Members field.										
Group Members	The left panel lists the items available to select.										
Selected Items	The right panel, Selected Items , lists the users and groups that are members of the group.										
Search	You can use the Search field to help you search for and quickly locate an item in the Group Members left panel. This is useful when the left panel contains more items than will display on a single page.										
LDAP Attributes	If the group is defined in an external directory service, this panel lists a set of attributes from the external directory service. The fields on this panel are based on the external directory service. LDAP attributes must be set by the system administrator. See “Exposing LDAP Attributes from an External Directory Service” on page 245 .										
Database Attributes	If the user is defined in an external database directory service, this panel lists a set of attributes from the external database directory service. The fields on this panel are based on the external database directory service. Database attributes										

Panel	Description
	must be set by the system administrator; see “Exposing Database Attributes from an External Directory Service” on page 246.

Managing Members of a Group

Members of a group can be users or other groups. You can add members to a group defined in the internal system directory service if they are also defined in the same directory service.

Note:

To work with users and groups defined in external directory services, use the mechanisms provided by the external directory services.

> To manage members in a group defined in the internal system service directory

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Groups**.
 - As system administrator: **Administration Dashboard > User Management > Manage Groups**.
- Search for the group you want to edit. For more information, see [“Searching for Existing Users, Groups, or Roles”](#) on page 142.
- In the search results, click any link in the row of the group you want to edit or click .
- Click **Group Members**.
- To manage the members of the group, do any of the following:
 - To add users (in the system directory service) to the group, move them to the **Selected Items** box.
 - To remove users from the group, move them from the **Selected Items** box.
 - To add groups (in the system directory service) to the group, move them to the **Selected Items** box.
 - To remove groups from the group, move them from the **Selected Items** box.
- When the **Selected Items** box lists all the members you want in the group, click **Save (Apply in system administration)**.

Making a Group a Member of Another Group

You can make a group a member of another group as long as both groups are defined in the internal system directory service. When one group becomes a member of a second group, all members of the first group also become members of the second group.

Note:

You cannot assign groups that are defined in an external directory service to a group defined in the internal system directory. Similarly, you cannot assign groups defined in the internal system directory service to an externally-defined group. You can, however, assign both internal and external groups to a role. See [“Managing Roles and Access to My webMethods” on page 183](#).

➤ To make a group a member of another group

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Groups**.
 - As system administrator: **Administration Dashboard > User Management > Manage Groups**.
- Search for the group you want to edit. For more information, see [“Managing External Directory Services” on page 112](#).
- In the search results, click any link in the row of the group you want to edit or click .
- Click **Groups**.
- To manage the membership of the group in other groups, do either of the following:
 - To make the current group a member of other groups (in the system directory service), move the parent groups to the **Selected Items** box.
 - To remove the current group as a member of other groups, move the parent groups from the **Selected Items** box.
- When the **Selected Items** box lists all the groups of which the current group should be a member, click **Save (Apply)** in system administration).

Deleting Groups

You can remove groups that you have previously defined in the internal system directory service.

Note:

When you delete a group, the definition for the group is removed, but the individual members of the deleted group (users and/or other groups) are not deleted.

➤ **To delete groups from the internal system directory service**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Groups.**
 - As system administrator: **Administration Dashboard > User Management > Manage Groups.**
2. Search for the groups you want to delete. For more information, see [“Searching for Existing Users, Groups, or Roles”](#) on page 142. Be sure to select **system** from the **Name** list.
3. In the search results, select the check boxes beside the names for the groups you want to delete, and click **Delete**.

11 Managing Permissions

■ Managing Permissions in My webMethods	170
■ Managing Access Privileges and Functional Privileges	173
■ Managing Permissions for an Individual Resource	174
■ Using Security Realms	176

Managing Permissions in My webMethods

The My webMethods Administrator manages permissions for the following resources:

- webMethods applications
- Tasks
- Workspaces

The System Administrator manages permissions for the following resources:

- webMethodsapplications
- Tasks
- Workspaces
- Content objects
- Portlet types
- Security realms

Both administrators can manage permissions for users, groups, and roles in any required combination.

Note:

Do not modify permissions for the built-in My webMethods Server roles from the Permissions Management page. Fixes you install to My webMethods Server subsequently might remove you changes.

Adding Permissions

The basic workflow in assigning permissions is to search for the resources on which to assign permissions.

> To assign permissions

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > Permissions Management.**
 - As system administrator: **Administration Dashboard > Configuration > Permissions Management.**
2. On the Search panel, from the **Resource Type** list, choose the resource type to be managed.

The following table lists the resources to which you can apply permissions:

Resource type	Description
webMethods Applications	webMethods applications that are installed in this instance of My webMethods.
Tasks	Tasks associated with webMethods Task Engine.
Workspaces	Workspaces that have been created on this instance of My webMethods.
Content Object	(System administrator only) Any resource on this instance of My webMethods Server, including files, folders, and pages.
Portlet Types	(System administrator only) Portlet types installed on this instance of My webMethods Server.
Security Realm	(System administrator only) Security realms created on this instance of My webMethods Server. For more information, see “Using Security Realms” on page 176 .

- If needed, apply a filter to narrow the search.

When you choose a Resource Type, a **FILTER** list appears with a defined set of filtering criteria. There are no filtering criteria for the webMethods Applications Resource Type because all installed applications are included.

- From the **FILTER** list, choose the filtering criteria.
For example, Workspace Name.
- Type a value to be searched for.
- If you need to add an additional filtering criterion, click  or  (depending on what type of administrator you are).

- Click **Search**.

The results of the search appear in the **Found** list.

- Move resources into the Selected list.

You can perform multiple searches to add resources to the list. Also, you can save searches to make it easier to locate the same resources again. For more information, see [“Working with Saved Searches” on page 144](#).

- On the Edit Permissions panel, click **Add Users/Groups/Roles**, search for Principals and move them to the **Selected** list.

You can perform multiple searches to add Principals to the list. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).

7. After all Principals have been selected, click **Add**.

The Permissions panel is displayed in a tree format containing the permissions that can be granted or denied for the Resource Type.

8. Click the **Grant** or **Deny** option for the settings, click **OK**, and then click **Apply**.

If neither option is selected, permissions for a setting will be determined from another source.

Modifying Permissions

You can modify previously set permissions to server resources. You can modify permissions for users, groups, or roles (Principals), or delete them entirely.

> To modify permissions

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > Permissions Management**.
 - As system administrator: **Administration Dashboard > Configuration > Permissions Management**.
2. If you have a saved search for the resources to be modified, do the following:
 - a. Click the Saved tab, choose the saved search from the **Saved Search** list and click Search
 - b. Move the result of the search to the **Selected** list and click **Next**.
3. Otherwise search for the resources by doing the following:
 - a. On the Search panel, from the **Resource Type** list, choose the resource to be managed.
 - b. If needed, apply a one or more filters to narrow the search.
 - c. Click **Search**.
 - d. Move resources into the Selected list.
 - e. Click **Next**.
4. To modify permissions for a Principal, do the following:
 - a. In the **Permissions** column of the Edit Permissions panel, click the link for the user, group, or role.

- b. Modify permissions for the various settings as needed, and click **OK**.
5. To delete permissions for a user, group, or role, select the checkbox in that row and click **Delete**.
6. Click **Apply**.

Managing Access Privileges and Functional Privileges

Permissions in My webMethods are grouped under access privileges and functional privileges.

Access privileges govern the rights of a user, group, or role to view applications and features in the Navigation panel, and access associated pages in My webMethods. Granting access privileges to a user also grants the user some functional privileges, and the user can both view and make changes to applications and features.

Functional privileges determine the rights of a user, group, or role to make changes within an application or feature, for example, to create and modify a workspace. Granting functional privileges requires granting access privileges.

The list of access privileges and functional privileges is determined by which webMethods applications are installed. Both the My webMethods administrator and the system administrator can grant or deny these privileges for users, groups, or roles.

Tip:

If you need to set the same set of privileges for multiple users, groups, and roles, you should consider aggregating all of them into a single role.

➤ To manage access privileges and functional privileges

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > Permissions Management**.
 - As system administrator: **Administration Dashboard > Configuration > Permissions Management**.
2. On the Search panel, from the **Resource Type** list, select **webMethods Applications**.
By default, My webMethods applications are already included in the **Selected** list.
3. Click **Next**.
4. On the **Manage Permissions** panel, click **Add**.
5. Search for users, groups or roles, move them to the **Selected** list, and click **Add**.

Note:

You can add only users or groups, or roles in this action. You cannot mix the three principals.

The Permissions panel displays a tree list, containing the permissions that can be granted or denied for Access Privileges and Functional Privileges.

6. In the tree list, select options as needed to assign Access Privileges and Functional Privileges.
 - Click **Grant** for each privilege that is to be granted.
 - Click **Deny** for each privilege that is to be denied.
 - Clear both options to neither explicitly grant or deny the privilege. Permissions will be determined from another source.
7. Click **OK** and then click **Apply**.

Managing Permissions for an Individual Resource

As system administrator you can manage individual resources within My webMethods Server. You can control the access of any Principal (user, group, or role) to a server resource or a hierarchy of server resources. Denying access to a server resource also prevents a Principal from seeing the resource in server navigation, such as in any listing of the resource's siblings or the contents of the resource's parent.

Note:

The information in this topic relates to managing permissions to pages and other server resources you develop that are not part of a webMethods application.

Viewing and Changing the Owner of a Server Resource

➤ To view and change the owner of a server resource

1. For a server resource, such as a page, click  or  and then click **Permissions**.

2. Click the Owner tab.

The Owner panel shows the owner of the server resource.

3. To change the resource owner, do the following:
 - a. In the Keywords field, type a user ID, click **Search**, move the user to the **Selected** box, and click **Apply**.
 - b. Click **Apply**.

Adding a Principal to the Permissions for a Server Resource

Remember that a Principal is a user, group, or role.

> To add a Principal to the permissions for a server resource

1. For a server resource, such as a page, click  or ) and then click **Permissions**.
2. On the Edit Permissions panel, click **Add Users/Groups/Roles**, search for Principals, and move them to the **Selected** list.

You can perform multiple searches to add Principals to the list. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).

3. After all Principals have been selected, click **Add**.

The Permissions panel is displayed in a tree format containing the permissions that can be granted or denied for the resource.

4. In the tree list, select options as needed.
 - Click **Grant** to explicitly grant permission.
 - Click **Deny** to explicitly deny permission.
 - Clear both options to neither explicitly grant nor deny permission. Permissions will be determined from another source.
5. Click **OK** and then click **Apply**.

Modifying Permissions for a Server Resource

> To modify the permissions for a server resource

1. For a server resource, such as a page, click  or , and then click **Permissions**.
2. To modify permissions for a Principal, do the following:
 - a. In the **Permissions** column of the Edit Permissions panel, click the link for the Principal.
 - b. Modify permissions for the various settings as needed, and click **OK**.
 - Click **Grant** to explicitly grant permission.
 - Click **Deny** to explicitly deny permission.

- Clear both options to neither explicitly grant nor deny permission. Permissions will be determined from another source.

3. Click **Apply**.

Removing a Principal from Server Resource Permissions

Remember that a Principal is a user, group, or role.

➤ **To remove a Principal from the permissions for a server resource**

1. For a server resource, such as a page, click  or , and then click **Permissions**.
2. On the Edit Permissions panel, select the checkbox for a Principal and click **Delete**.
3. Click **Apply**.

Using Security Realms

As a system administrator, when you manage permissions, as described in “[Managing Permissions for an Individual Resource](#)” on page 174, you do so one resource at a time. This method is satisfactory for small servers, but can be cumbersome as the number of pages and users increases. Security realms enable system administrators to manage permissions for resources based on users, groups, or roles, making it easier to manage large servers. After a Security realm is applied to a resource, individually set permissions do not apply unless you specifically choose to set them.

For convenience, you can organize Security realms into folders called *containers*. The following table lists the Security realm containers that My webMethods Server provides by default:

Container	Security realms
Forum Realms	Security Realms that manage permissions for forums.
My webMethods Security Realms	Security Realms that manage permissions for My webMethods resources.
Portal Resources	Security Realms that manage permissions for My webMethods Server resources

You can create, rename, and remove containers on the **Security Realms Administration** page.

There are several default security realms that manage permissions for My webMethods Server resources, all of which reside in the Portal Resources container. System administrators have the right to read, modify, or delete the server resources. Other users have permissions, described in the following table:

Security Realm	Managed Permissions
Administrative Commands	Administrative resources. Permits only server administrators to read, modify, or delete the resource.
Directory Management Commands	Resources that manage users, group, and roles. Permits users to view or execute the resource. Anonymous users are denied access.
Directory Service Commands	Resources that manage directory services. Permits users to view or execute the pages. Anonymous users are denied access.
Portal Developer Commands	Resources for the development and maintenance of pages and content. Members of the Developers group are granted the right to read, modify, or delete the resource. Permits users to view or execute the resource. Anonymous users are denied access.
Public Commands	Resources for interacting with a server, such as logging in. Permits all users, including anonymous users, to read or execute the resource, but not to modify or delete it.
Restricted Commands	Resources for interacting with a server after one has logged in. Permits users who have logged in to read or execute the resource, but not to modify or delete it. Anonymous users are denied access.
User Profile Management Commands	Resources that control the look and feel of the server. Permits users to view or execute the resource. Anonymous users are denied access.

You create and manage Security Realms that you can use for your content on the Security Realms Administration page.

Creating a Container

You can create a container at the same level as the default containers or you can create a container within a container.

» To create a new container

1. As system administrator, click **Administration > Configuration > Security Realms Administration**.
2. On the **Security Realms Administration** page, do one of the following:
 - To create a new container at the current level, click the **Create New Container** tab.
 - To create a nested container, click the name of an existing container, and then click the **Create New Container** tab.
3. In the **Name** field of the **Create New Container** tab, type a display name for the container.

4. (Optional) In the **Description** field, type a description.
5. Click **Create Container**.

Removing a Container

Important:

If you remove a container, any Security Realms or other containers within it are also removed.

> To remove a container

1. As system administrator, click **Administration Dashboard > Configuration > Security Realms Administration**.
2. Click  for the container you want to remove, and then click **Remove Container**.

Renaming a Container

To rename a container, do the following:

> To rename a container

1. As system administrator, click **Administration Dashboard > Configuration > Security Realms Administration**.
2. Click  for the container you want to rename, and then click **Modify Container**.
3. In the **Name** field, type the new name.
4. (Optional) In the **Description** field, type a new description.
5. Click **Update**.

Creating a Security Realm

> To create a new Security Realm

1. As system administrator, click **Administration Dashboard > Configuration > Security Realms Administration**.
2. Click the name of the container in which to create the Security Realm.

Note:

After you have created a Security Realm within a container, you cannot move it to another container.

3. Click **Create New Security Realm**.
4. In the **Name** field, type a display name for the Security Realm.
5. (Optional) In the **Description** field, type a description.
6. (Optional) To add an extended type, move it to the **Selected Items** box and click **Select**.
7. (Optional) To add an external policy provider, move it to the **Selected Items** box.
8. Click **Create Security Realm**.
9. Click  for the new Security Realm, and then click **Configure Permissions**.
10. On the Edit Permissions panel, click **Add Users/Groups/Roles**, search for Principals, and move them to the **Selected** list.

You can perform multiple searches to add Principals to the list. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).

11. After all Principals have been selected, click **Add**.

The Permissions panel is displayed in a tree format containing the permissions that can be granted or denied for the Security Realm.

12. In the tree list, select options as needed.
 - Click **Grant** to explicitly grant permission.
 - Click **Deny** to explicitly deny permission.
 - Clear both options to neither explicitly grant nor deny permission. Permissions will be determined from another source.
13. Click **OK** and then click **Apply**.

Removing a Security Realm

To remove a Security Realm, do the following:

- **To remove a Security Realm**

1. As system administrator, click **Administration Dashboard > Configuration > Security Realms Administration**.
2. Click the name of the container in which the Security Realm resides.
3. Click  for the Security Realm you want to remove, and then click **Remove Security Realm**.

Renaming a Security Realm

To rename a Security Realm, do the following:

> To rename a Security Realm

1. As system administrator, click **Administration Dashboard > Configuration > Security Realms Administration**.
2. Click the name of the container in which the security realm resides.
3. Click  for the security realm you want to rename, and then click **Modify Security Realm**.
4. In the **Name** field, type the new name.
5. (Optional) in the **Description** field, type a new description.
6. Click **Update**.

Adding Resources to a Security Realm

To add resources to a Security Realm, do the following:

> To add resources to a Security Realm

1. As system administrator, click **Administration Dashboard > Configuration > Security Realms Administration**.
2. Click the name of the container in which the security realm resides.
3. Click  for the security realm you want to manage, and then click **Manage Objects**.
4. On the **Manage Security Realm** panel, click **Add Portal Resource**.
5. In the left panel, browse to a server resource to be added to the security realm.

6. To have the security realm manage a server resource, move the resource to the **Selected Items** box and click **Add Items**.

Removing Resources from a Security Realm

To remove resources from a Security Realm, do the following:

> To remove resources from a Security Realm

1. As system administrator, click **Administration Dashboard > Configuration > Security Realms Administration**.
2. Click the name of the container in which the security realm resides.
3. Click  for the security realm you want to manage, and then click **Manage Objects**.
4. For the resource you want to remove, Click  and then click **Remove**.

12 Managing Roles and Access to My webMethods

■ About Roles in My webMethods Server	184
■ Granting Users Access to My webMethods and the My webMethods Users Role	185
■ Creating Roles	186
■ Editing Information for a Role	192
■ Deleting Roles	193
■ Defining Dynamic Attributes Associated with a Role	194

About Roles in My webMethods Server

A *role* is a collection of users, groups, or other roles. A set of default roles is installed with My webMethods Server. You can add users, groups, and other roles to this initial set.

The following table lists the default roles, available in My webMethods Server and the resources these roles can access:

Default role	Description
Admin Role	Provides access to all My webMethods Server resources. By default, the SysAdmin and Designer users are members of this role.
My webMethods Administrators	Provides access to user management and other functions needed by the My webMethods Administrator, who is a default member of this role.
My webMethods Users	Provides access to the My webMethods user interface for all users of My webMethods applications. By default, the My webMethods Server Administrator is a member of this role, but you must add all other users to it. For more information, see “Granting Users Access to My webMethods and the My webMethods Users Role” on page 185.

The members assigned to a role can span across multiple directory services. That is, their membership can include users, groups, and roles defined in the internal directory service, as well as, users and groups defined in external directory services. The membership of a role can be static, like groups where each member is specifically assigned. However, you can also make the membership of a role dynamic. It is valid for roles to be recursive, making it possible for roles to be members of each other.

The following table lists the different ways you can define the membership of a role, whether the membership is static or dynamic, and where to find more information about how to define membership for each type of role:

Defining Membership	Static or Dynamic	Additional Information
Specify the users, groups, and other roles you want to be members of the role	Static	“Adding a Static Role” on page 186
Specify an LDAP query that queries an external directory service to determine the users or groups assigned to the role	Dynamic	“Adding an LDAP Query Role” on page 188
Specify the criteria for a rule that My webMethods Server executes to determine the users and groups assigned to the role	Dynamic	“Adding a Rule-Based Role” on page 188

Defining Membership	Static or Dynamic	Additional Information
Specify a database query that queries a database directory service to determine the users or groups assigned to the role	Dynamic	“Adding a Database Role” on page 191

You can associate dynamic attributes with a role to provide more information about a role. For example, if there is a “Customer Service role, an administrator might add a “Location attribute to identify where the user assigned to the “Customer Service role is located. When you add the attribute to the role, you assign it a value. This becomes the default value for the attribute. Continuing with the example, assume your main service center is in Ohio. As a result, when you add the “Location attribute to the “Customer Service role, you assign it the value `Ohio`. You can also assign user-specific values to a dynamic attribute. Once again, continuing with the example, assume you have a user that is a member of the “Customer Service role, but who is located in Colorado rather than Ohio. You can assign that user a specific value of `Colorado` for the “Location attribute in the “Customer Service role.

For more information about dynamic attributes, see:

- [“Setting User-Specific Values for Dynamic Attributes” on page 195](#)
- [“Deleting Dynamic Attributes Assigned to a Role” on page 196](#)

Granting Users Access to My webMethods and the My webMethods Users Role

My webMethods Server includes the My webMethods Users role, which is already defined for you. The My webMethods Users role governs access to My webMethods. That is, a user *must* be a member of the My webMethods Users role for the system to allow the user to log into the user interface. The My webMethods Users role is a static role; that is, the membership of the My webMethods Users role is a specified list of users, groups, and other roles.

Users Defined in the Internal System Directory Service

When you add users to the internal system directory service, My webMethods Server automatically assigns the new users to the My webMethods Users role. You do not need to take any further action.

Externally-Defined Users

You can configure My webMethods Server to obtain information about users and groups from external directory services. After configuring an external directory service, you need to take further action to identify the externally-defined users that you want to be able to access My webMethods. These users must be identified in the My webMethods Users role. For information about using external directory services with My webMethods Server, see [“Configuring an External LDAP, ADSI, or ADAM Directory Service” on page 112](#) and [“Allowing Externally Defined Users to Perform Actions from My webMethods” on page 125](#).

Creating Roles

My webMethods Server administrators can create different types of user roles. My webMethods Server uses all roles, regardless of type, in the same manner, but role membership is identified differently for each type.

The following table describes the types of roles you can create and how to identify role membership for each role type:

Role	Identifying Membership
Static Role	Specify a collection of users, group, and roles that are members of the role you are creating. The membership of the role does not change unless you manually edit the role and change its membership. For more information, see “Adding a Static Role” on page 186.
	<p>Note: This is similar to a group except that role membership can span multiple directory services.</p>
LDAP Query Role	Specify an LDAP query. The users, groups, and roles that match the query become members of the role. The membership of the role is dynamic based on the outcome of the query at run time. For more information, see “Adding an LDAP Query Role” on page 188.
Rule Based Role	Specify a rule that My webMethods Server executes to determine the membership. The users, groups, and roles that match the rule become members of the role. The membership of the role is dynamic based on the outcome of the execution of the rule at run time. For more information, see “Adding a Rule-Based Role” on page 188 .
Database Role	Specify a query for a database directory service. The users, groups, and roles that match the query become members of the role. The membership of the role is dynamic based on the outcome of the query at run time. For more information, see “Adding a Database Role” on page 191 .

Adding a Static Role

A static role is a simple collection of users, groups, and other roles.

➤ To create a static role

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles > Add Role.**

- As system administrator: **Administration Dashboard > User Management > Manage Roles > Add Role.**
2. In the **Role Name** field, type the name that you want to assign to the new role.
Valid role names can contain only letters, numbers, an underscore, or a space character.
 3. To select the **Static Role Provider**, move that role provider to the **Selected Items** box.
 4. Click **Create Role**.

Editing Members of a Static Role

To edit the members of a static role, do the following:

> To edit members in a static role

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles**
2. Search for the static role to which you want to add members. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142.](#)
3. In the search results, click the role name or click .
4. On the Members panel, click **Edit Members**.
5. To add members, do the following:
 - a. Under **Search For**, choose the **Users, Groups, or Roles** option.
 - b. In the **Keywords** field, type a keyword representing the users, groups, or roles you want to search for, and click **Search**.
 - c. Move one or more users, groups, or roles to the **Selected** box.
6. To delete members from the static role, move them from the **Selected** box.
7. Click **Apply**.

Adding an LDAP Query Role

An LDAP query role is based on an LDAP query to an external directory service. Any user or group that meets the requirements of the query is a member of the role.

> To create an LDAP query role

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles > Add Role.**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles > Add Role.**
2. In the **Role Name** field, type the name you want to assign to the new role.

Valid role names can contain only letters, numbers, an underscore, or a space character.
3. To select the **LDAP Query Role Provider**, move that role provider to the **Selected Items** box.
4. Click **Create Role**.
5. In the **LDAP Query** field type a valid LDAP query.
6. Select the **Simple Query** option if the query in the **LDAP Query** field contains simplified LDAP query syntax.

Unless you are creating a complex LDAP query, the query syntax can be cumbersome to use. With the **Simple Query** option, the syntax is filled in for you. For example, to find all persons whose manager has the user ID abrown, the simple query syntax is `manager=abrown`.
7. In **LDAP Directory Service**, click **Browse**.
8. Move the LDAP directory service to the **Selected Items** box and click **Select**.
9. In the **Principal Type** list, choose whether the query searches for **Users** or **Groups**.
10. To update the LDAP query, click **Save (Apply** in system administration).

Adding a Rule-Based Role

A rule-based role is based on a server rule. Any user, group, or role that matches the rule is a member of the role.

> To create a rule-based role

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles > Add Role.**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles > Add Role.**
2. In the **Role Name** field, type the name you want to assign to the new role.

Valid role names can contain only letters, numbers, an underscore, or a space character.
3. To select the **Rule Based Role Provider**, move that role provider to the **Selected Items** box.
4. Click **Create Role**.
5. Under the **Match Criteria** heading, select **Match All Criteria Below** or **Match Any Criteria Below** as the criteria for the rule-based role.
6. Fill in the appropriate match criteria for the rule-based role using the following guidelines:

User DN Value(s): A regular expression that matches any part of the current user's directory distinguished name (DN). In the field, type the portions of the DN to which you want a match.

For example, `ou=Engineering.*ou=US` matches a user with the following DN:

```
uid=joe,ou=Development,ou=Engineering,ou=Midwest,ou=US,o=webMethods
```

Domain Name Expression: A regular expression that matches any part of the name of the current user's directory service as registered in My webMethods Server. In the field, type the directory service name to which you want a match.

For example, `US` (without quotes) matches a user from the US Corporate directory service. This is a very effective way to govern the look and feel for users that may be in different user directories, such as partners.

Group DN and Role DN Expression: A regular expression that matches any part of any group or role of which the current user is a member. In the field, type the portions of the DN to which you want a match.

For example, `ou=Engineering` matches a user belonging to a group with the following DN:

```
cn=portal,ou=Engineering,ou=Midwest,ou=US,o=webMethods.
```

User Attributes: One or more pairs of user attributes and their values from the user's record. If you have more than one user attribute, the value set in **Match Criteria** determines how attributes are matched:

- **Match All Criteria Below** - Each regular expression must match some part of the corresponding attribute value for the current user.

- **Match Any Criteria Below** - Any regular expression in the list can match some part of the corresponding attribute value for the current user.

For example, if the rule is configured to match all criteria, and the configured user attribute pairs are listed in the following table:

Name	Value
office	Bellevue
telephonenumber	(425) 564-0000

and the current user's attribute values are listed in the following table:

Name	Value (current user)
office	Bellevue
telephonenumber	(206) 123-4567

the rule does not match the current user because it matches the `office` attribute value but not the `telephonenumber` attribute value. If, however, the rule is configured to match any criteria, the preceding example rule does match the current user.

To create an attribute-value pair, click **Add**. At the prompt, type the attribute name and click **OK**. At the prompt, type the value to be matched and click **OK**.

Request Headers: One or more pairs of HTTP header attributes and values. You can match anything that appears within an HTTP header, such as the browser agent string or the kinds of MIME types the user will accept. The rule can be a regular expression, or a simple text string. If you have more than attribute-value pair, the value set in **Match Criteria** determines how attributes are matched:

- **Match All Criteria Below** - Each regular expression must match some part of the corresponding attribute value for the request header.
- **Match Any Criteria Below** - Any regular expression in the list must match some part of the corresponding attribute value for the request header.

For example, if the rule is configured to match all criteria, and the configured request header pairs are listed in the following table:

Name	Value
Accept-Charset	utf-8
Accept-Language	ja

and the request header values for the current user are listed in the following table:

Name	Value (current user)
Accept-Charset	ISO-8859-1,utf-8;q=0.7
Accept-Language	en-us,en;q=0.5

the rule does not match the current user because it matches the Accept-Charset header value but not the Accept-Language header value. If, however, the rule was configured to match any criteria, the rule does match the current user.

To create an attribute-value pair, click **Add**. At the prompt, type the attribute name and click **OK**. At the prompt, type the value to be matched and click **OK**.

Parent Resource: A resource that matches the current resource or a parent of the current resource. To select a resource, click **Browse** to open the resource selector and select a resource against which to match the rule. If you want match a resource that is referenced by an alias, you can optionally click **Use Alias** to select an existing alias on My webMethods Server.

Resource Type: A resource type that matches the current resource type. To select a resource type, click **Browse** to open the resource selector and select a resource type, from the Extended Types folder, against which to match the rule. If you want match a resource type that is referenced by an alias, you can optionally click **Use Alias** to select an existing alias on My webMethods Server.

Resource Property: One or more pairs of resource properties and values. If you know the internal name of a property associated with a resource, you can match it. If you have more than one property-value pair, the value set in **Match Criteria** determines how properties are matched:

- **Match All Criteria Below** - Each regular expression must match some part of the corresponding attribute value for the request header.
- **Match Any Criteria Below** - Any regular expression in the list must match some part of the corresponding attribute value for the request header.

For example, if you want to match files that are PDFs, the property-attribute pair is `mimeType=pdf`.

To create an property-value pair, click **Add**. At the prompt, type the attribute name and click **OK**. At the prompt, type the value to be matched and click **OK**.

7. Click **Apply**.

Adding a Database Role

A database role is based on a query to a database directory service. Any user, group, or role that matches the rule is a member of the role.

Note:

To create a database role, you must first connect to the database as an external data source. See [“Managing External Data Sources” on page 129](#).

➤ To create a database role

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles > Add Role**.
 - As system administrator: **Administration Dashboard > User Management > Manage Roles > Add Role**.
- In the **Role Name** field, type the name that you want to assign to the new role.
Valid role names can contain only letters, numbers, an underscore, or a space character.
- To select the **Database Role Provider**, move that role provider to the **Selected Items** box.
- Click **Create Role**.
- From the **Datasource** list, select the database to be used as a data store.
- If the role can include users, in the **Query User** field, type a SQL query that returns a record for a given user in the database who should be a member of the role.

The parameters to the query are:

- {uid}—Principal unique ID
- {dn}—Principal distinguished name

An example of a valid query is:

```
select * from user-roles where roleID='Admin' and userid='{uid}'
```

- If the role can include groups, in the **Query Group** field, type a SQL query that returns a record for a given group in the database that should be a member of the role.
- Click **Save**.

Editing Information for a Role

To update the information for a role, perform the following procedure:

➤ To edit a role

- To navigate to the correct page, do one of the following:

- In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles**
2. Search for the role that you want to edit. For more information, see [“Searching for Existing Users, Groups, or Roles”](#) on page 142.
 3. In the search results, click the role name or click  .
 4. On each panel, edit the role information as required and click **Save (Apply** in system administration).

Important:

My webMethods Server displays the information grouped on various panels. After making changes to information on a single panel, be sure to click **Save (Apply** in system administration) to save your changes *before* selecting another panel.

The following table lists the panels displayed for a role:

Panel	Additional Information
Role Information	“Adding a Static Role” on page 186
	“Adding an LDAP Query Role” on page 188
	“Adding a Database Role” on page 191
Dynamic Attributes	“Defining Dynamic Attributes Associated with a Role” on page 194

Deleting Roles

If you no longer need a role, you can delete it.

Note:

When you delete a role, the members of the role (users, groups, and/or other roles) are not deleted.

> To delete a role

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles**

2. Search for the roles you want to delete. For more information, see [“Searching for Existing Users, Groups, or Roles”](#) on page 142.
3. In the search results, select the check boxes beside the roles you want to delete, and click **Delete**.

Defining Dynamic Attributes Associated with a Role

You can associate dynamic attributes with a role. Some webMethods applications use dynamic attributes.

When you define the attribute in the role, you assign it a value. The value you assign to the role is considered the default value. The default value is used for all users unless a user-specific value is defined for the attribute.

» To define a dynamic attribute for a role

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles**
2. Search for the role. For more information, see [“Searching for Existing Users, Groups, or Roles”](#) on page 142.
3. In the search results, click the role name or click .
4. Click **Dynamic Attributes**.
5. Click **Add Attribute**.
6. On the **Add an Attribute** page, specify the following:

The following table lists the properties to configure when adding a dynamic role attribute:

Field	Description
Attribute Name	The internal name for the dynamic attribute. There are no restrictions on what characters you can use for this name.
Display Name	The name you want My webMethods Server to use when it displays information about this attribute in the user interface.
Data Type	The data type of the attribute. Select a data type from the list.

Field	Description
Value	The value you want assigned to the attribute. This is the default value for the attribute. This value will be used for all users assigned to this role unless a user-specific value is defined for the attribute. For more information, see “Setting User-Specific Values for Dynamic Attributes” on page 195 .

7. Click **Save**.

Setting User-Specific Values for Dynamic Attributes

The default value assigned to a dynamic attribute is the default for all users unless you assign a user-specific value for the attribute. To assign a user-specific value to a dynamic attribute, perform the following procedure.

➤ To assign a user-specific value to a dynamic attribute

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Users**.
 - As system administrator: **Administration Dashboard > User Management > Manage Users**.
2. Search for the user for which you want to assign a user-specific value to a dynamic attribute. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).
3. In the search results, click any link in the row of the user or click .
4. Click **Roles**.
5. In the **Role Member Attributes** panel, locate the dynamic attribute for which you want to assign a user-specific value, and type the value in the **User Value** column.

Note that in the display the attributes are grouped by role; the role name appears before the list of attributes assigned to the role.

6. Click **Save(Apply** in system administration).

Changing the Order of Dynamic Attributes Assigned to a Role

You can change the order in which dynamic attributes display for a role. Changing the order affects only how the system displays the attributes in the **Role Member Attributes** panel for a

user, but does not otherwise affect how the attributes are used by My webMethods Server or by webMethods applications.

➤ **To change the order of dynamic attributes assigned to a role**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles**
2. Search for the role. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).
3. In the search results, click the role name or click .
4. Click **Dynamic Attributes** and then click **Change Attribute Order**.
5. To reorder dynamic attributes, move them up or down as needed.
6. When the dynamic attributes are in the order you want, click **Save (Apply** in system administration).

Deleting Dynamic Attributes Assigned to a Role

If you no longer want one or more dynamic attributes assigned to a role, you can delete them.

➤ **To delete one or more dynamic attributes assigned to a role**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > User Management > Roles**
 - As system administrator: **Administration Dashboard > User Management > Manage Roles**
2. Search for the role. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).
3. In the search results, click the role name or click .
4. Click **Dynamic Attributes**.

5. Select the check boxes beside the dynamic attributes you want to delete, and click **Delete Selected Attributes**.
6. Click **Save**.

13 My webMethods Server Clustering

■	How a My webMethods Server Cluster Works	200
■	Planning Your My webMethods Server Cluster	204
■	Considerations When Building a My webMethods Server Cluster	206
■	Modifying the Cluster Configuration	207
■	Modifying Resource Locking Settings for a Cluster	217
■	Configuring the Number of Purge Threads for a Cluster	218
■	Monitoring and Controlling Your Cluster	218
■	Removing a Component from a Cluster	222
■	Working with the cluster.xml File	222
■	Creating a Cluster Node from an Image	225
■	Partitioning Applications on Cluster Nodes	227

How a My webMethods Server Cluster Works

A My webMethods Server cluster is an active/active environment in which multiple server instances run at the same time, sharing the My webMethods Server database. A My webMethods Server cluster achieves high scalability by distributing the workload among multiple servers. This model is different from an active/passive environment, which makes use of a standby server. A My webMethods Server cluster achieves high availability through the use of shared resources, allowing the cluster to continue to function when a node is taken out of service.

To create a cluster, you need to install an instance of My webMethods Server on each machine in the cluster. All nodes of the cluster share the same My webMethods Server database, which contains shared configuration information and data, stored by the system content service.

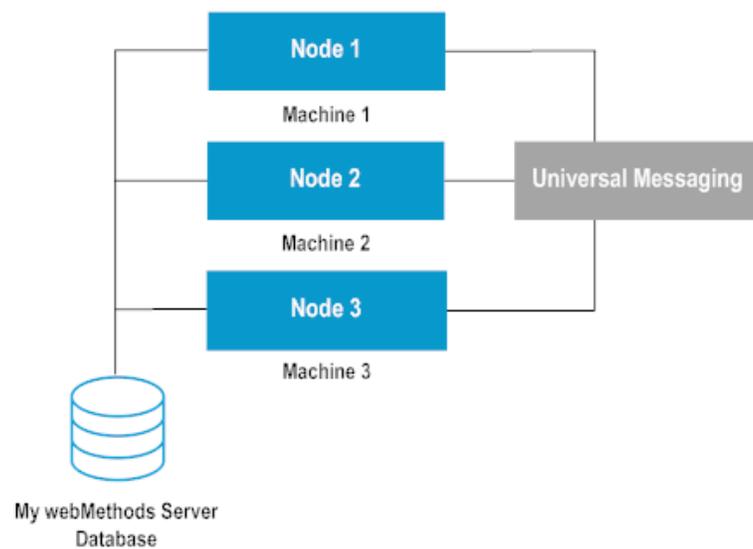
The system content service stores and retrieves files and objects published by webMethods users and applications, running on My webMethods Server. The system content service is installed with My webMethods Server and does not require additional configuration. Published content is available to all nodes in the cluster.

You should install the same set of webMethods applications (such as Task Engine), My webMethods user interfaces, and language packs on each node in the cluster. If an application is not installed on a particular node, the components of the application are not functional on that node.

The nodes in the cluster exchange information when they run, using the following channels:

- The My webMethods Server database - cluster bookkeeping is maintained this way.
- HTTP - when a cluster node delegates the execution of a command to another node, it uses the HTTP port of the target server.
- The Universal Messaging server - My webMethods Server nodes use the Universal Messaging server to exchange JMS events for cache synchronization between the nodes.

Cluster architecture



The Front End URL

In a production environment, all users and all webMethods applications that communicate with the My webMethods Server cluster need to use the Front End URL established for the cluster, described in [“Setting the Front End URL” on page 207](#). In addition, a production cluster should include the use of a load balancer or external web server, as shown in [“High Availability in a My webMethods Server Cluster” on page 201](#).

In practice, users log into the Front End URL through the load balancer. In turn, the load balancer distributes calls to the nodes. A cluster can have only one Front End URL.

Note:

The load balancer must be set up to use sticky sessions. A user session, once established, is routed to the same server machine until the session is closed.

When an HTTP request is issued to produce a My webMethods Server page, My webMethods Server uses the actual host name and port number of that request, no matter how the request is routed: directly or indirectly through the load balancer or web server. In these cases a Front End URL is not necessary.

For several use cases, like generating My webMethods Server URLs from within email notifications or creating links to tasks from within Task Engine, Front End URL configuration is required because no HTTP request is available at that time.

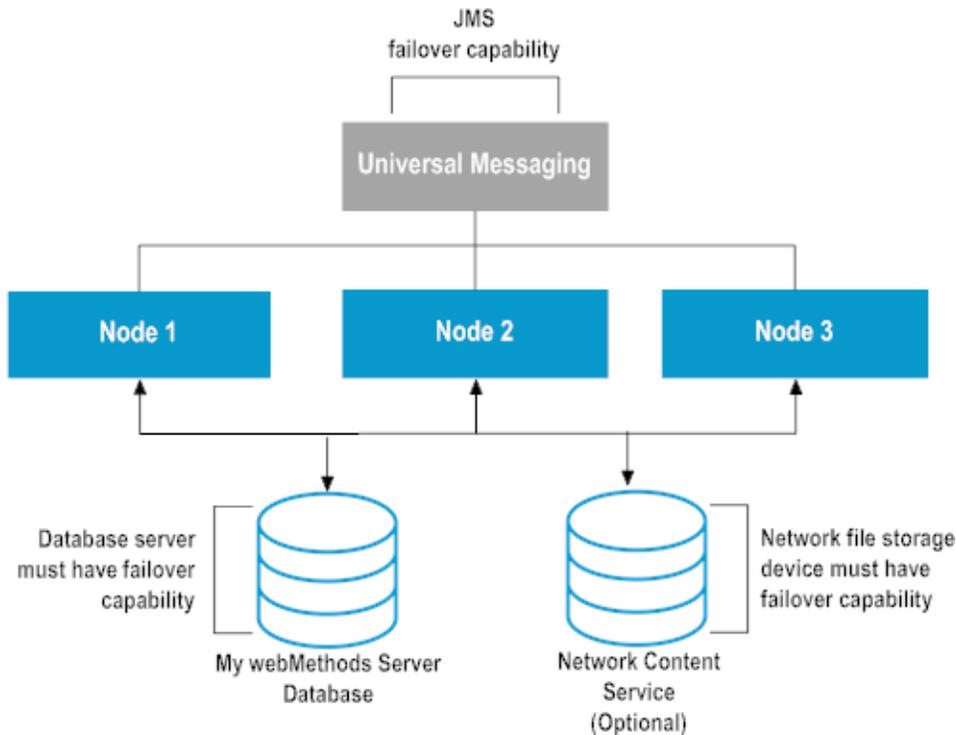
High Availability in a My webMethods Server Cluster

By itself, a My webMethods Server cluster does not provide high availability. You must also ensure the availability of data stored by the nodes in the cluster. To provide high availability, the database server used by the My webMethods Server database must have a failover capability.

If the cluster uses the default system content service, it in turn uses the My webMethods Server database, so it is covered by the database server failover capability. However, if you use a network content service, the network file storage device must have failover capability to support cluster high availability.

Any description of the configuration of third-party failover devices is beyond the scope of this guide.

Requirements for a highly available My webMethods Server cluster



Component Deployment in a My webMethods Server Cluster

A My webMethods Server cluster uses an asymmetrical mode of deploying components such as portlets or deployable packages.

In Asymmetrical component deployment mode, a component deployed to a node is not automatically deployed to other nodes in the cluster. In Asymmetrical deployment mode, you need to manually deploy a component locally to each node where it is needed. In this mode it is possible to have different versions of a component running at the same time on different nodes of a cluster. Because component registration is centralized in the My webMethods Server database, the registration is performed only once by the first node on which the component is deployed. Because of this dependency, changes made to a component that can affect component registration or My webMethods Server taxonomy can potentially break other versions of the component residing on a different node. Some examples of changes to a component that can cause changes to the My webMethods Server database schema are:

- Adding or removing a portlet preference.
- Adding, removing, or changing a field in a DBO (Dynamic Business Object) table.
- Adding, removing, or changing a field in a task type business data.
- Adding, removing, or updating My webMethods Server taxonomy

Asymmetrical component deployment mode enables you to deploy and test a modified component on one or more nodes before deploying it to the entire cluster.

Cluster Roles

There are several cluster roles that can be assigned to a node in a cluster. These include:

- The **Auto Deploy** cluster role allows for the automatic deployment of portlets that are copied to the *Software AG_directory \MWS\server\serverName\deploy* directory (of a node that has the Auto Deploy role). A cluster can have multiple Auto Deploy nodes. This role is enabled by default, but is only needed if you want the automatic deployment capability. It is often desirable to have this role disabled in a production environment to reduce the possibility of unauthorized modification of the server. This role does not affect cluster high availability.
- The **Notification** cluster role is responsible for formatting and sending email notifications, and notifications that are sent to a user. A cluster can have multiple Notification nodes. The Notification role is enabled by default, but is needed only if you use notifications in your production environment. For example, if you are running Task Engine, users will not receive task notifications if the Notification role is not enabled. For high availability it is recommended that you have more than one cluster node in this role. If all Notifications nodes go down, notification request messages continue to accumulate in the queue. No notifications are generated or delivered until one or more Notification nodes comes back online and starts processing them.
- The **Search** role is responsible for indexing all content that is exposed to the embedded search engine, maintaining the search index, and performing the searches. The Search role is enabled by default, and should be enabled on all cluster nodes where the search functionality will be needed. The Search role is important to server operation and there are core features that will not operate without it.
 - Permissions Management
 - Workspace Management and Add to Workspace

Each cluster node maintains a local copy of the search index, improving the performance and reliability of searches. If the search index becomes corrupted on one cluster node, you can remove that node from the load balancer while the search index is rebuilt on that node. In the meantime, the other nodes can continue servicing requests.

- The **Task engine** cluster role is responsible for all Task Engine activities, such as queuing tasks, processing task rules, searching and retrieving task data. A cluster can have multiple Task Engine roles. This role is enabled by default, but is needed only if you actually have Task Engine running on your cluster. For high availability it is recommended that all nodes have this role enabled unless a specific node is not included in the load balancer configuration and never services end-user or Process Engine requests.

Guidelines for Assigning Specific Cluster Roles

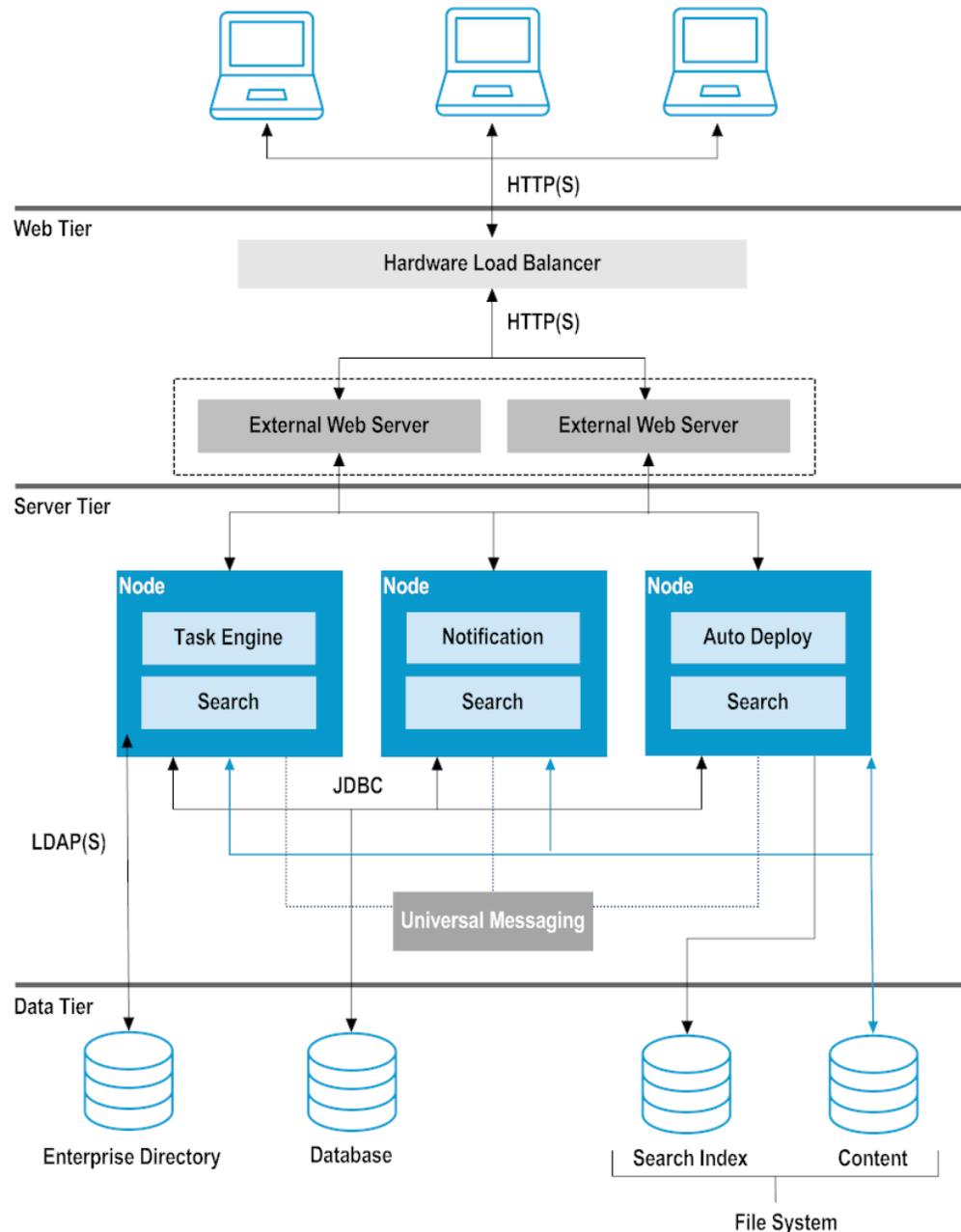
You can assign cluster roles to each node in a cluster following the guidelines that follow. Any time you change the roles assigned to a cluster node, you need to restart the node for the assignments to take effect.

- The Auto Deploy, Task Engine, and Notification roles can be assigned to as many nodes as you need.
- Nodes that are running one or more roles can be further broken down and separately clustered. This may be necessary if your scalability requirements warrant setting up additional cluster roles to handle increased traffic.
- The configuration data for each node is stored in the cluster.xml file. It is possible to edit the data for your cluster directly in the cluster.xml file, provided you know the proper data structures and values for the properties of your cluster. For more information, see [“Working with the cluster.xml File” on page 222](#).
- Any changes to the cluster configuration will require each node to be restarted.

Planning Your My webMethods Server Cluster

The following diagram shows just one of many types of distributed deployments that can be configured with My webMethods Server.

A My webMethods Server Cluster Network Diagram



The preceding diagram assumes that there will be three server machines operating in a cluster, each with its own cluster roles. The machines defined in the web tier include a hardware-based load balancer and two web server machines. The following sections illustrate the process that an administrator would work through to build a cluster.

The procedure for configuring Apache in an external load-balanced web server cluster configuration with your deployment is explained in [“Using My webMethods Server with Web Servers”](#) on page 43.

Considerations When Building a My webMethods Server Cluster

Consider the following information when building a My webMethods Server cluster:

- All servers in the cluster must use *the same* My webMethods Server database. Make sure that you use the same My webMethods Server database JDBC URL when you install each node in the cluster.
- You should install *the same* set of webMethods applications (such as Task Engine), My webMethods User Interfaces, and language packs on each node in the cluster.
- To build a My webMethods Server cluster, install each node in the cluster following the instructions provided in *Installing Software AG Products*. You must provide the following information:
 - The HTTP port number for each server instance. The default value is 8585. If you install two server nodes on the same machine, you must use a different HTTP port for each node.
 - The JDBC URL for the My webMethods Server database. The URL must include any additional properties needed to support database server failover.
 - The URL of the Universal Messaging server or Universal Messaging cluster to use as a JMS provider. My webMethods Server nodes go in maintenance mode if you do not specify a Universal Messaging URL, or the Universal Messaging server is unavailable. If you do not specify a Universal Messaging URL during installation, you must log in as Administrator and configure the URL on the **Cluster Administration** page in My webMethods.
- You can start the nodes in the cluster concurrently or one after the other, depending on your needs. The first time you start each node, it performs bootstrap activities and loads shared configuration files from the My webMethods Server database. The node is automatically added to the cluster. You can also add nodes to a cluster manually, as described in [“Adding a Node to a Cluster” on page 211](#).

Important:

Do not make any cluster configuration changes while you are starting cluster nodes for the first time. If you do so, you may lose the configuration of the nodes that perform self-configuration automatically when starting for the first time.

If the startup of a node completes successfully, you can open a browser window and log into My webMethods Server as system administrator to verify that the node is running correctly.

- If you are building a production cluster, perform the following steps:
 - If you know you will need a network content service, set it up as described in [“Configuring a New Content Service” on page 341](#).
 - If you already have a load balancer or external web server set up, change the Front End URL to point there as described in [“Setting the Front End URL” on page 207](#).

Modifying the Cluster Configuration

You can perform several functions to modify the configuration of a standalone server, a cluster, or a node in a cluster.

Setting the Front End URL

All users and all webMethods applications that communicate with the My webMethods Server cluster must use the Front End URL established for the cluster. By default, this value is the URL of the first node of the cluster. The URL can point to a load balancer or external web server. A cluster can have only one Front End URL.

➤ To change the Front End URL for a cluster

1. Navigate to the correct page by doing one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Advanced or Clustered Configuration.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Advanced or Clustered Configuration.**
2. In the **MwS Front End URL** field, type a fully qualified URL:
 - `http://host_name:port_number`
 - OR
 - `https://host_name:port_number` if you want to access the cluster using only a secure connection,

where *port_number* must have the same value as the **MWS Front End Secure Port** field.
3. To access secure My webMethods Server content, in the **MWS Front End Secure Port** field, type an HTTPS port number.

The default value is -1, which means that no secure Front End URL is configured for the cluster.
4. Click **Submit**.
5. Restart the My webMethods Server cluster.

Configuring the Connection to the Universal Messaging Server

By default, My webMethods Server uses the Software AG Universal Messaging server as a Java Message Service (JMS) provider. However, the connection to the Universal Messaging server is not configured by default. When you start a node in a My webMethods Server cluster for the first

time, the node starts in maintenance mode. To exit maintenance mode, you must login to the cluster node, and configure the connection to the Universal Messaging server. When the other nodes in the cluster are accessible over the network, My webMethods Server automatically propagates the configuration. If My webMethods Server does not update the configuration of a node automatically, log in to the node and reload the cluster configuration from the Cluster Administration page.

Note:

A node in the cluster enters maintenance mode when the node loses connection to the Universal Messaging server, and exits maintenance mode when the server becomes available.

> To configure the connection to the Universal Messaging server

1. Navigate to the following page:

- As My webMethods administrator: **Navigate > Applications > Administration > My webMethods > Cluster Settings.**
- As sysadmin: **Administration Dashboard > Configuration > Cluster Administration.**

2. On the **Advanced or Clustered Configuration** tab, specify the location of the Universal Messaging server in the **JNDI Provider URL** field.

The default URL of the Universal Messaging server in a local installation is:

```
nsp://localhost:9000
```

3. Click **Submit**.

Configuring Secure Connection to the Universal Messaging Server

You configure secure connection to Universal Messaging on the **Advanced or Clustered Configuration** tab of the **Cluster Settings** page in My webMethods, separately for each node in the My webMethods Server cluster.

> To configure SSL connection to the Universal Messaging server:

1. Log on to the cluster node as My webMethods Administrator.
2. Navigate to **Applications > Administration > My webMethods > Cluster Settings > Advanced or Clustered Configuration.**
3. On the **Advanced or Clustered Configuration** tab, click **Advanced Settings.**
4. In the **SSL settings** dialog, specify the following:

The following table lists the fields to configure to connect to Universal Messaging using SSL:

Field	Description
Keystore path	Fully qualified path to the location of the keystore for client certificates.
Keystore password	The password for the specified keystore.
Keystore certificate name	Required when client certificate validation is enabled in Universal Messaging. The alias of the certificate in the specified keystore to use for authentication. For more information about client certificate validation, see the Universal Messaging documentation.
Truststore path	Fully qualified path to the location of the truststore for CA certificates.
Truststore password	The password for the specified truststore.
SSL protocol	Optional. The type of protocol to use for secure connection. The default value is TLS.
Cipher suite	Optional. A comma-separated list of the names of the cipher suites to use for encryption. For more information about available cipher suites, see the Universal Messaging documentation.

5. Click **Submit**.

My webMethods Server stores the SSL settings for the current cluster node in the profile configuration for the node. If you configure Universal Messaging SSL in the `custom_wrapper.conf` file for My webMethods Server, changing the SSL configuration in My webMethods will override the settings in the `custom_wrapper.conf` file.

Alternatively, you can configure the SSL connection to Universal Messaging using the `custom_wrapper.conf` file for My webMethods Server. For information about how to modify SSL settings in the `custom_wrapper.conf` file, and how to configure SSL connection to Universal Messaging for Common Directory Services (CDS), see [“Configuring Secure Connection to the Universal Messaging Server for CDS” on page 209](#).

Configuring Secure Connection to the Universal Messaging Server for CDS

You configure SSL connection between a My webMethods Server cluster node and the Universal Messaging server by specifying additional JVM properties in the `custom_wrapper.conf` file for My webMethods Server. When you configure an SSL connection to Universal Messaging for a clustered installation that includes Integration Server and Common Directory Services (CDS), you must also add these properties in the `custom_wrapper.conf` file for the Integration Server that uses CDS.

For more information about the `custom_wrapper.conf` file, see [“Configuring JVM Settings for My webMethods Server” on page 95](#). For more information about editing the `custom_wrapper.conf` file for Integration Server, see *webMethods Integration Server Administrator’s Guide*.

The following table lists the JVM properties that you add to configure secure connection to the Universal Messaging server for CDS:

Property	Description
<code>nirvana.ssl.keystore.path</code>	Fully qualified path to the location of the keystore for client certificates.
<code>nirvana.ssl.keystore.pass</code>	The password for the specified keystore.
<code>nirvana.ssl.keystore.cert</code>	Required when client certificate validation is enabled in Universal Messaging. The alias of the certificate in the specified keystore to use for authentication.
<code>nirvana.ssl.truststore.path</code>	Fully qualified path to the location of the truststore for CA certificates.
<code>nirvana.ssl.truststore.pass</code>	The password for the specified truststore.
<code>nirvana.ssl.protocol</code>	The type of protocol to use for secure connection. If you do not specify a protocol for secure connection, My webMethods Server and CDS use TLS.

Configuring Credentials to Connect to the Universal Messaging Server

By default, My webMethods Server and CDS use the `mwsJMSAdmin` user to connect to the Universal Messaging server. Certain Universal Messaging security settings might require explicitly granting the `mwsJMSAdmin` user permissions for the messaging realm.

You can configure a specific My webMethods Server user to connect to Universal Messaging by setting the following system properties in the `custom_wrapper.conf` file for My webMethods Server, or the `custom_wrapper.conf` file for the Integration Server instance that hosts CDS:

The following table lists the system properties for configuring credentials for connecting to Universal Messaging.

Parameter	Description
<code>lax.nl.env.PRINCIPAL</code>	The name of the user to connect to the Universal Messaging server.
<code>lax.nl.env.PASSWORD</code>	The password of the user to authenticate to the Universal Messaging server.

The user must be configured with proper access permissions for the Universal Messaging server. For more information about the Universal Messaging security settings, see the Universal Messaging documentation.

For more information about working with the `custom_wrapper.conf` file for My webMethods Server, see [“The Java Service Wrapper” on page 92](#).

For more information about working with the `custom_wrapper.conf` file for Integration Server, see *webMethods Integration Server Administrator’s Guide*.

The My webMethods Server Cluster ID

Multiple My webMethods Server clusters, or individual nodes can use the same Universal Messaging Server. To identify that a node belongs to a cluster, My webMethods Server generates a unique identifier for the cluster in the **MWS Cluster ID** field. My webMethods Server appends the **MWS Cluster ID** to JMS topics and queues to distinguish the messages, intended for a cluster when using Universal Messaging as a JMS provider. You cannot edit the **MWS Cluster ID** field.

Adding a Node to a Cluster

A node is automatically added to the cluster when you start it for the first time. The new node must have the same My webMethods Server database as the other nodes in the cluster and should have the same set of webMethods applications (such as Task Engine), My webMethods User Interfaces, and language packs on each node in the cluster.

➤ To add a node to a cluster

1. Install the cluster node following instructions provided in *Installing Software AG Products*. You need to provide the following information:
 - The HTTP port number for this server instance. The default value is 8585.
 - The JDBC URL for the My webMethods Server database used by the cluster. This URL should include any additional properties needed to support database server failover.
2. Start the My webMethods Server instance.

The first time you start an additional node, it performs bootstrap activities and loads shared configuration files from the My webMethods Server database. The node is automatically added to the cluster.

3. After the new node initializes, reload the cluster configuration on the **Cluster Administration** page, or restart the cluster, as described in [“Restarting or Stopping All Nodes in a Cluster” on page 220](#).

Modifying a Node in a Cluster

You can modify several properties for a node in a cluster or a standalone instance of My webMethods Server. Common reasons to modify a node are to add security by specifying an HTTPS port, add support for an external web server, or change cluster roles. You can also modify the properties of a cluster node or a standalone server manually. See [“Working with the cluster.xml File” on page 222](#).

➤ **To modify a standalone server or a node in a cluster**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Advanced or Clustered Configuration.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Advanced or Clustered Configuration.**
2. For a standalone server or a node in the cluster, do any of the following.
 - In the **HOST** field, change the host name (including domain if appropriate) or IP address of the machine where the node is to run.

The specified host name or IP address must resolve to the correct machine running the node. It must be valid for all nodes in the cluster. See your network administrator for information about host names and network setup at your site.
 - In the **HTTP PORT** field, change the port number to be used by the HTTP listener.

This field must always have a valid port number.
 - In the **HTTPS PORT** field, type the port number to be used by the HTTPS listener.

A value of 0 (zero) in this field disables the listener.

Note:
My webMethods Server includes a sample HTTP certificate that you can use to set up and test your HTTPS listener. The sample is located in the *Software AG_directory* \MWS\server\serverName\config\localhost.p12 file (see [“Certificates Used for Secure Connections” on page 34](#)). For production environments, be sure to obtain an actual Certificate from a qualified authority such as Verisign.

 - To select one or more roles for a node, select the check box for the role.
3. Click **Submit**.
4. Restart the cluster or standalone server for the changes to take effect, as described in [“Restarting or Stopping All Nodes in a Cluster” on page 220](#) or [“Restarting or Stopping Individual Nodes in a Cluster” on page 219](#).

Assigning a Search Role to a Node

The Search role is enabled by default. You can assign the Search role to nodes on the cluster as needed.

➤ **To assign the Search role to a node**

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Advanced or Clustered Configuration.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Advanced or Clustered Configuration.**
2. Select the **Search** option for nodes that are to be assigned the role and clear the option for nodes that should not have the Search role.
3. Restart the cluster or standalone server for the changes to take effect, as described in [“Restarting or Stopping All Nodes in a Cluster” on page 220](#) or [“Restarting or Stopping Individual Nodes in a Cluster” on page 219](#).

Renaming a Node

Nodes in a cluster are automatically assigned node names when you add them to the cluster. My webMethods Server displays the name of a cluster node in the **Name** column of the **Advanced or Clustered Configuration** page, and in the **Node Name** column of the **Cluster Status and Control** in My webMethods.

You can change the name of the cluster node using the `mws update` command, as follows:

> To rename a node in a cluster

1. At a command line prompt on the machine that hosts the node, move to the `bin` directory for My webMethods Server:

```
Software AG_directory \MWS\bin
```

2. Change the node name by running the following command:

```
mws -s instanceName update -Dnode.name=newNodeName
```

where *instanceName* is the name of the server instance that corresponds to the node and *newNodeName* is the new name of the cluster node. The instance name remains unchanged.

The cluster node name is updated in the following files:

- `cluster.xml`
- `Software AG_directory \MWS\server\serverName\config\cluster.node.properties`
- `Software AG_directory \MWS\server\serverName\config\cluster.node.properties.bak`
- `Software AG_directory \profiles\MWS_serverName\configuration\custom_wrapper.conf`

Deleting a Node from a Cluster

When a node in a My webMethods Server cluster is no longer in use, you should delete it from the cluster by doing the following:

➤ To delete a node from a cluster

1. From another node, stop the node to be deleted from the cluster, as described in [“Restarting or Stopping Individual Nodes in a Cluster”](#) on page 219.
2. To retrieve the cluster.xml file from the My webMethods Server database, type this command:

```
mws getconfig cluster.xml
```

3. Open the cluster.xml file in a text editor, which you can find at this location:

```
Software AG_directory \MWS\server\serverName\config
```

4. In the cluster.xml file, locate the node to be deleted, remove the entire <Component> element for this node, and save the file. For example:

```
<Component class="com.webmethods.portal.system.cluster.impl.Server"
  enabled="true" name="nodeName">
  .
  .
  .
</Component>
```

5. To deploy the revised cluster.xml file to the My webMethods Server database, type this command:

```
mws putconfig cluster.xml
```

6. Delete the cluster.xml file from the `\serverName\config` directory.
7. On the **Cluster Administration** page, reload the cluster configuration, or restart the cluster, as described in [“Restarting or Stopping All Nodes in a Cluster”](#) on page 220.

Uninstalling a Node

If you uninstall a My webMethods Server instance that is a node in a cluster, the node is automatically deleted from the cluster configuration. If the My webMethods Server database is not available at the time of uninstallation, you must delete the node manually as described in [“Deleting a Node from a Cluster”](#) on page 214.

Modifying Database Connection Retries

If a server loses its connection with the My webMethods Server database, it tries to reestablish the connection. This would be the case if a database failover occurs in a high availability environment. You can modify the number of times the server retries the connection and the interval it waits between retries. If you have multiple servers in a cluster, you need to modify these values individually on each machine. Also, if you have multiple server instances on the same machine, you need to modify the values for each server instance.

My webMethods Server database connection information for each server instance is maintained in the `mws.db.xml` file found in this location:

Software AG_directory \MWS\server\serverName\config

The following table lists the available settings for database connection retries:

Value	Description
Retry count	<p>The number of times My webMethods Server attempts to re-execute a SQL statement in the case of connection loss, deadlock, or any other SQL error not normally expected. If the value is:</p> <ul style="list-style-type: none"> ■ 0—the server does not try to reestablish the connection. ■ An integer of 1 or more—the server retries the connection that number of times. <p>Typically, you should set the retry count to no more than three. If the server needs more than that number, there are problems with the connection or with the database server.</p>
Retry delay	<p>The time in milliseconds the server waits between retries. The retry delay value should be large enough to allow for a failover to occur, and is dependent on the configuration of your database server.</p>

To modify retry behavior for a single My webMethods Server database connection, do the following.

➤ To modify database retry behavior for a server instance

1. Open the `mws.db.xml` file, which you will find at the following location, in a text editor:

Software AG_directory \MWS\server\serverName\config

2. To change the number of retries, for each node in the cluster, add the `<RETRYCOUNT>` element to `mws.db.xml` within the `<PARAMS></PARAMS>` tags:

```
<RETRYCOUNT>2</RETRYCOUNT>
```

3. To change the number of milliseconds to delay between retries, for each node in the cluster, add the `<RETRYDELAY>` element to `mws.db.xml` within the `<PARAMS></PARAMS>` tags:

```
<RETRYDELAY>300</RETRYDELAY>
```

The combined elements might look like this:

```
<PARAMS>
.
.
.
<!-- retry count on error -->
<RETRYCOUNT>2</RETRYCOUNT>
<!-- delay in ms between retries -->
<RETRYDELAY>300</RETRYDELAY>
</PARAMS>
```

4. Save the file and restart the My webMethods Server instance.
5. Repeat this procedure for each node in the cluster.

Configuring My webMethods Server to use the database for JMS communication

You can use the database as a JMS provider for communication between My webMethods Server and the Common Directory Services component in Integration Server, or for My webMethods Server clusters that exchange a small number of JMS events. For more information, see [“Considerations when using the database as a JMS provider” on page 216](#).

» To configure My webMethods Server to use the database as a JMS provider

1. Retrieve the cluster.xml file from the My webMethods Server database using the `getConfig` command.
2. On the `<Cluster>` element, insert the `useDbJms="true"` attribute, as follows:

```
<Cluster useDbJms="true" clusterId="706790370
```

3. Save the file and execute the `putConfig` command to deploy the file to the database.
4. Delete the local copy of the cluster.xml file from the `\serverName\config` directory.

If you do not delete the cluster.xml file, this node will continue to use the local version of the file. For more information about working with the cluster.xml file, see [“Working with the cluster.xml File” on page 222](#).

Considerations when using the database as a JMS provider

My webMethods Server clusters and CDS require the use of a JMS provider for synchronization between the nodes. The default JSM provider for synchronization between My webMethods Server nodes, and between My webMethods Server and Integration Server when using CDS, is Universal Messaging.

You can configure My webMethods Server to use the database as a JMS provider for cluster synchronization, as described in [“Configuring My webMethods Server to use the database for JMS communication” on page 216](#). The following considerations apply when using the database as a JMS provider:

- My webMethods Server uses the production database for cluster messaging. You cannot configure a separate datasource for JMS.
- When the cluster nodes exchange a large number of events, the load on the production database increases. This might affect overall cluster performance.
- My webMethods Server does not provide JMS failover capability, and relies on the failover capabilities of the JMS provider.

Modifying Resource Locking Settings for a Cluster

When installing additional components in a My webMethods Server cluster, cluster nodes obtain database locks on the resources that get updated. Component installation might require a particular resource to remain locked for a significant amount of time. To ensure that concurrent attempts to lock a resource between cluster nodes do not result in timeouts, you can configure the lock validity and wait times, using the following additional JVM parameters in the `custom_wrapper.conf` file for My webMethods Server:

```
com.webmethods.task.lock.max.time
com.webmethods.task.lock.max.wait.time
```

➤ To modify the resource locking settings for a My webMethods Server cluster

1. Shut down My webMethods Server.
2. Open the `custom_wrapper.conf` file in a text editor.

You can find the file at the following location:

```
Software AG_directory \profiles\MWS_serverName\configuration
```

3. In the `custom_wrapper.conf` file, specify the time limits for resource locking and lock waiting by adding the following additional parameters:

The following table lists the additional JVM parameters that configure resource locking for a cluster:

Parameter	Description
<code>-Dcom.webmethods.task.lock.max.wait.time=time_in_milliseconds</code>	The maximum amount of time in milliseconds for which a node can wait to acquire a lock on a resource. The default value is 300000 (5 minutes).

Parameter	Description
<code>-Dcom.webmethods.task.lock.max.time=time_in_milliseconds</code>	The maximum amount of time in milliseconds for which a node can lock a resource. The default value is 300000 (5 minutes).

4. Save the `custom_wrapper.conf` file and restart My webMethods Server.

For more information about how to set additional JVM parameters in the `custom_wrapper.conf` file, see [“Configuring JVM Settings for My webMethods Server” on page 95](#).

Configuring the Number of Purge Threads for a Cluster

When a user deletes an item, My webMethods Server marks the item for deletion. The item remains in the system until My webMethods Server purges all pending items, according to a pre-configured schedule, or as a result of a manual trigger of the content purge by a system administrator.

When a purge operation runs in a My webMethods Server cluster, all cluster nodes that have the autodeploy role attempt to purge the pending items. If a large number of items is marked for deletion, the purge operation might execute for a significant amount of time. You can configure the number of threads that remove deleted items from the system and speed up the purge operation using the `purge.schedule.maxthreads` additional JVM property. The default value of the property is 10.

➤ To configure the number of purge executor threads

1. Open the `custom_wrapper.conf` file in a text editor.
2. In the `custom_wrapper.conf`, file, add the following additional JVM property:

```
purge.schedule.maxthreads=number_of_threads
```

3. Save the file and restart My webMethods Server.

For more information about working with the `custom_wrapper.conf` file, see [“Configuring JVM Settings for My webMethods Server” on page 95](#).

Monitoring and Controlling Your Cluster

You can see status information about a node in a cluster or a standalone server, restart or stop an individual node, restart or stop an entire cluster, or place an individual node in maintenance mode.

Checking Status Information About a Cluster

You can update status information for the nodes in a cluster or a standalone server manually.

➤ To see status information about nodes in a cluster

- To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Cluster Status and Control.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Cluster Status and Control.**
- To update the status information on the **Status and Control** page, click **Refresh**.

The following table lists the fields that the **Status and Control** page displays:

Field	Description
	Indicator of whether the node is running or not. Click the icon for a node to update individual status. Click  to update status for all nodes. The status icons are: <ul style="list-style-type: none">  The node is running  The node is stopped
NODE NAME	The node name assigned to the node. If you click the node name, the browser connects directly to the node using the current window. You can change this name manually, as described in “Renaming a Node” on page 213 .
ROLES	The cluster roles currently assigned to the node.
UP TIME	The time since the node was last started in days, hours, and minutes.
ACTIVE USERS	The number of active users for the node. This number may not equal the exact number of logged-in users because it can include guest sessions and abandoned sessions.
ACTIONS	Actions you can take on the node, as described in “Restarting or Stopping Individual Nodes in a Cluster” on page 219 .

Restarting or Stopping Individual Nodes in a Cluster

You can restart or stop an individual node in a cluster.

➤ To restart or stop individual nodes in a cluster

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Cluster Status and Control.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Cluster Status and Control.**
2. For the node to be acted upon, do either of the following:
 - Click **Restart.**

The node is stopped, causing the  icon to be displayed. After the node is restarted, the  icon returns.
 - Click **Shutdown.**

The node is stopped. After you have stopped the node, you cannot restart it from this page. You need to start the My webMethods Server instance on the host machine.

Restarting or Stopping All Nodes in a Cluster

You can restart or stop all nodes in a cluster with a single command.

➤ To restart or stop all nodes in a cluster

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Cluster Status and Control.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Cluster Status and Control.**
2. Do either of the following:
 - Click **Restart Cluster.**

The nodes are stopped and restarted one at a time beginning with the first node in the table. If the first node is the one from which you issue this command, the restart will begin with the next node. Your node is restarted after all other nodes have restarted. The page does not automatically refresh after the node is restarted. You should manually refresh the browser after a minute or so. The login screen is displayed when the clustered is fully restarted.
 - Click **Shutdown Cluster.**

All nodes are stopped. To restart the nodes, you need to start the My webMethods Server instance on each host machine.

Placing a Cluster Node in Maintenance Mode

You can place an individual node in a cluster in maintenance mode to perform operations such as diagnosing an issue, cleaning up the file system, or changing the server configuration.

When My webMethods Server is in maintenance mode, the behavior of the server differs depending on the type of user:

- **My webMethods and system administrators.** My webMethods administrators and system administrators can log in and interact normally with My webMethods Server while the server is in maintenance mode.
- **My webMethods users.** My webMethods users cannot log in while the server is in maintenance mode. A My webMethods user who is already logged in will be redirected to the maintenance mode page when the user attempts to load another server page.

➤ To place an individual node in maintenance mode

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Cluster Status and Control.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Cluster Status and Control.**
2. For the node that you want to place in maintenance mode, click **Maintenance** in the Actions column.
3. In the Maintenance dialog box, specify values for the following fields:
 - a. In the **In Maintenance Mode** field, click **Yes**.
 - b. In the **Duration** field, select one of the following values:
 - **Persistent** - The node remains in maintenance mode when My webMethods Server is restarted.
 - **Temporary** - The node exits maintenance mode when My webMethods Server is restarted.
 - c. (Optional) In the **Reason** field, specify a reason for placing the node in maintenance mode. If you do not specify a reason, My webMethods Server displays a default maintenance message on the maintenance page.
4. Click **Apply**.

On the **Cluster Status and Control** tab of the Cluster Settings page, My webMethods Server displays **Yes** in the In Maintenance column for the node.

Taking a Cluster Node out of Maintenance Mode

> To take an individual node out of maintenance mode

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Cluster Status and Control.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Cluster Status and Control.**
2. For the node that you want to take out of maintenance mode, click **Maintenance** in the Actions column.
3. In the **In Maintenance Mode** field, click **No**.
4. Click **Apply**.

On the **Cluster Status and Control** tab of the Cluster Settings page, My webMethods Server displays **No** in the In Maintenance column for the node.

Removing a Component from a Cluster

Because all nodes in a cluster use the same My webMethods Server database, registration information for all deployed components is shared among the nodes. If you remove a component from one node, the registration is removed from the database and the component is no longer available on any node.

> To remove a component from a cluster

- Remove the component file from each node's deploy folder (if each node has the Autodeploy role enabled)
—OR—
- Use Deployer to roll back the component on each node in the cluster

Working with the cluster.xml File

The cluster.xml file contains configuration information for a standalone server or for all servers in a cluster. This file resides in the My webMethods Server database. My webMethods Server adds new servers to this file when you add nodes to a cluster, and modifies it when you make changes in the Cluster Administration page. You can also modify the cluster.xml file manually, as shown in [“Renaming a Node” on page 213](#) and [“Deleting a Node from a Cluster” on page 214](#).

The following fragment shows a basic configuration for a node in a cluster:

```
<Component class="com.webmethods.portal.system.cluster.impl.Server"
  enabled="true"
  name="server-one-node59581">
  <Properties host="server-one" name="http port="8585"/>
  <Properties host="server-one" name="https" port="0"/>
  <Role name="autodeploy"/>
  <Role name="taskengine"/>
  <Role name="notification"/>
</Component>
```

The following table lists the properties, included in the sample fragment and describes their sample values:

Property	Example value
Node name	server-one-node59581
Host name	server-one
HTTP port	8585
HTTPS port	0—Indicates the HTTPS listener is disabled.
Roles	Indicates the cluster roles supported by this server are autodeploy, taskengine, and notification.

The following table lists the tasks that you perform by modifying the cluster.xml file:

Action	Description
Add a node	Add a complete <Component> element to the file, as shown in the preceding example. In this case, the node must already exist and must use the same My webMethods Server database as other nodes in the cluster. The node name in the cluster.xml file must agree with the node name in the custom_wrapper.conf file on the host machine for that node, or must be specified in the command line used to start the node: <code>mws -n nodeName start</code>
Rename a node	Change the node name of a node.
Modify node attributes	Edit any of the attributes for a node, including node name, HTTP, and HTTPS ports, and cluster roles.
Delete a node	Remove the complete <Component> element that defines the node.
Modify cluster configuration	Edit the cluster properties, described in the cluster properties table.

The following table lists the cluster properties, you can configure using the cluster.xml file:

Property	Description
frontEndUrl	The URL for the My webMethods Server cluster. Specify a fully qualified URL, for example: <code>http://host_name:port_number</code> , or <code>https://host_name:port_number</code> . The default value is the URL of the first node of the cluster.
frontEndSecurePort	The HTTPS port that the My webMethods Server cluster uses for secure communication. The default value is <code>-1</code> , which means that no secure port is configured.

Editing the cluster.xml File

To edit the cluster.xml file a My webMethods Server instance does not have to be running, but the My webMethods Server database does.

> To edit the cluster.xml file

1. At a command line prompt on any machine that hosts a cluster node, move to the bin directory of the server instance:

```
Software AG_directory \MWS\bin
```

2. To retrieve the cluster.xml file from the My webMethods Server database, type this command:

```
mws getconfig cluster.xml
```

3. Open the cluster.xml file in a text editor.

You can find the file at this location:

```
Software AG_directory \MWS\server\serverName\config
```

4. Make a backup copy of the cluster.xml file.
5. Modify the contents of the cluster.xml file as needed.
6. To deploy the revised cluster.xml file to the My webMethods Server database, at the command line prompt type this command:

```
mws putconfig cluster.xml
```

7. Delete the cluster.xml file from the `\serverName\config` directory.

If you do not delete the cluster.xml file, this node will continue to use the local version of the file.

- Restart the cluster, as described in [“Restarting or Stopping All Nodes in a Cluster”](#) on page 220.

Backing Out of a Change to the cluster.xml File

If you use the Cluster Administration page to make a change to a cluster.xml file that introduces an error, you can return to the previous configuration using the cluster.xml.bak file. A My webMethods Server instance does not have to be running, but the My webMethods Server database does.

> To back out of a change to the cluster.xml file

- At a command line prompt on any machine that hosts a cluster node, move to the bin directory of the server instance:

```
Software AG_directory \MWS\bin
```

- To retrieve the cluster.xml.bak file from the My webMethods Server database, type this command:

```
mws getconfig cluster.xml.bak
```

- Change the name of cluster.xml.bak to cluster.xml.
- To deploy the revised cluster.xml file to the My webMethods Server database, at the command line prompt type this command:

```
mws putconfig cluster.xml
```

- Delete the cluster.xml file from the `\serverName\config` directory.

If you do not delete the cluster.xml file, this node will continue to use the local version of the file.

- Restart the cluster, as described in [“Restarting or Stopping All Nodes in a Cluster”](#) on page 220.

Creating a Cluster Node from an Image

One method to create a cluster is to save an image of a machine with an installed My webMethods Server instance and other products, and move it to a new machine. This effort saves installation time and results in an identical configuration on the new machine. It is recommended that you create an image of a fully initialized My webMethods Server instance instead of just a My webMethods Server installation. The My webMethods Server instance becomes fully initialized after it has been started for the first time and the administrator successfully logs in and then stops it.

Before you can use the new node in a cluster, you need to make a few changes. The new machine needs to have access to the My webMethods Server database, but you should not start My webMethods Server before making the following changes.

➤ **To add a node created from an image**

1. On the machine with a new image, at a command line prompt, move to the bin directory of the server instance:

```
Software AG_directory \MWS\bin
```

2. To retrieve the cluster.xml file from the My webMethods Server database, type this command:

```
mws getconfig cluster.xml
```

3. Open the cluster.xml file in a text editor.

You can find the file at this location:

```
Software AG_directory \MWS\server\serverName\config
```

4. Copy the entire <Component> element for the server instance on the original machine and paste it back into the cluster.xml file.
5. For the new <Component> element, make these changes:

- a. Change the `name` value to a node name descriptive of the new node.

- b. Change the `host` property to host name of the new machine

For example:

```
name="server-one-node59581"><Properties
```

```
host="server-one" name="http"
```

might be changed to this:

```
name="node2"><Properties
```

```
host="server-two" name="http"
```

- c. As needed, change the <Role> values to reflect the cluster roles to be assumed by the new node.

6. To deploy the revised cluster.xml file to the My webMethods Server database, at the command line prompt type this command:

```
mws putconfig cluster.xml
```

7. Delete the cluster.xml file from the `\serverName\config` directory.

If you do not delete the cluster.xml file, this node will continue to use the local version of the file.

8. Open the custom_wrapper.conf file for the server instance in a text editor. You can find the file at this location:

```
Software AG_directory \profiles\MWS_serverName\configuration\
```

9. In the custom_wrapper.conf file, change the value of the NODE_NAME statement and save the file:

```
set.NODE_NAME=nodeName
```

10. Start the My webMethods Server instance.
11. If the startup completes successfully, open a browser window and log into My webMethods Server as SysAdmin, to verify it is running correctly.
12. Restart each node in the cluster.

You can restart nodes manually, or you can do so from within My webMethods Server, as described in [“Restarting or Stopping All Nodes in a Cluster” on page 220](#).

Partitioning Applications on Cluster Nodes

In a My webMethods Server cluster, *partitioning* is the division of applications among nodes of the cluster. By configuring cluster partitions, you can control which applications run on a node and which do not run. Partitioning is not the same as asymmetrical mode ([“Component Deployment in a My webMethods Server Cluster” on page 202](#)), which causes components to be deployed to a node manually. Partitioning controls the actual execution of components.

Why Would I Want Partitions?

There are several practical reasons why you might want to create partitions in your cluster:

- Separation of clients

Through the use of load balancer rules, you can route one set of users to one node and another set of users to a different node. Users on a partition have access to only the set of applications you choose to provide for them.

- Better control over application management

Because you run certain applications only within one partition, it is easier to update an application while the rest of the cluster continues to operate normally.

- Cache stability and memory usage

If a particular node experiences slow performance due to heavy traffic, users on other nodes are not affected. In the current Task Engine architecture for example, tasks typically reside in

an in-memory cache. The existence of too many task instances in the cache will overrun the available memory.

Guidelines for Partitions

There are a few things you should know about how partitions are created within a cluster:

- By default, a cluster has a single partition. If there is only one partition, all nodes are part of it, regardless of whether you have explicitly listed them as part of the partition.
- Each node is part of a partition. Any nodes you have not specifically listed as part of a partition become members of the last partition defined in the configuration files. See [“Creating and Modifying Partitions” on page 228](#).
- Each partition is associated with a `phaseProvider.xml` file, which governs whether portlets are enabled or disabled. If a partition is not associated with a specific phase provider file, it is associated with the default phase provider by default.

Creating and Modifying Partitions

By default, there is one partition that includes all cluster nodes, as described in [“The Default Partition Configuration” on page 231](#). To create or modify additional partitions, you need to modify the partition configuration files.

My webMethods Server uses configuration files to manage partitioning on a cluster.

- `clusterPartitions.xml`—Modify this file to add partitions to the cluster and assign nodes to each partition. See [“The Cluster Partition File” on page 229](#).
- `partitionNamePhaseProvider.xml`—Create and modify these files to specify whether portlets are enabled or disabled. See [“The Phase Provider File” on page 229](#).

Each partition can have its own phase provider file. A partition that does not have a unique phase provider associated with it uses the default `phaseProvider.xml` file.

- `partitionNamePortlets.properties`—Create and modify one of these files for each phase provider. This file lists the portlets affected by the phase provider for a specific partition. See [“The Portlets Properties File” on page 230](#).

Each partition can have its own portlets properties file. If all portlets in the partition are enabled, a portlets properties file is not required.

Note:

File names and file contents are case sensitive.

My webMethods Server stores these configuration files in the My webMethods Server database. To get access to these files so you can modify them, see [“Modifying Configuration Files Stored in the Database” on page 100](#).

The Cluster Partition File

The cluster partition file, `clusterPartitions.xml`, defines the partitions on the cluster and assigns nodes to each partition. The data structure for a partition in the `clusterPartitions.xml` file looks like this:

```
<ClusterPartitions>
  <Partition name="partitionName"
    frontEndUrl="http://some.url.com">
    <Component name="componentName">
    </Component>
  </Partition>
</ClusterPartitions>
```

where:

The table describes the expected values of configurable attributes in the `clusterPartitions.xml` file.

<i>partitionName</i>	Is the name assigned to the partition. Other configuration files use this name to identify the partition.
<i>some.url.com</i>	An alternate Front End URL. The <code>frontEndUrl</code> attribute is optional, allowing the partition to use a different Front End URL than the cluster as a whole. If you omit this attribute, the partition uses the <code>frontEndUrl</code> value in the <code>cluster.xml</code> file.
<i>componentName</i>	The name of a component (a cluster node) that is part of the partition. The component name is the same as the component name in the <code>cluster.xml</code> file.

Entries in `clusterPartitions.xml` conform to the following rules:

- There is one `Partition` element for each partition in the cluster. There is always at least one `Partition` element.
- Each `Component` element represents a cluster node that is a member of the partition. There can be zero or more nodes specified for a partition.
- If there is only one `Partition` element in the file, all cluster nodes are members of that partition, regardless of whether they are represented by a `Component` element.
- Any cluster node not specifically represented within a `Partition` element is automatically a member of the last partition defined in the file.

The Phase Provider File

A phase provider file, `partitionNamePhaseProvider.xml`, determines whether portlets are enabled or disabled for a partition. Each partition can have its own phase provider file. A partition that does not have a unique phase provider associated with it uses the default `phaseProvider.xml` file. The portion of the phase provider file that applies to this topic is in the `portlet` phase nested inside the `CoreServices` phase:

```

<!-- All services come on line -->
<Phase name="CoreServices" enabled="true"
  class="com.webmethods.portal.system.init.impl.DefaultPhase">
  .
  .
  <PhaseInfo name="portlet" enabled="true"
    class="com.webmethods.portal.service.portlet.impl.PortletProvider"
    initFile="config:/adminPortlets.properties"
    initFileComponentsEnabled="false"
  />
  .
  .
/>

```

where:

The following table describes the expected values of configurable attributes in the phase provider file.

<code>initFile</code>	Specifies the portlets properties file containing a list of portlets to be affected by the phase provider for the partition.
<code>initFileComponentsEnabled</code>	Specifies whether the portlets listed in the partition portlets properties file are to be enabled or disabled: <ul style="list-style-type: none"> ■ <code>false</code>: The listed portlets will be disabled. ■ <code>true</code>: The listed portlets are the only ones that will be enabled.

In the example above, the portlets properties file is `adminPortlets.properties` and the `initFileComponentsEnabled` attribute is `false`, meaning that file contains a list of portlets to be disabled.

The `initFile` and `initFileComponentsEnabled` attributes are not required. If you omit these attributes, all portlets are enabled in the partition.

Note:

To create a new phase provider file for a partition, make a copy of the default `phaseProvider.xml` file and make changes as needed.

The Portlets Properties File

The portlets properties file, *partitionNamePortlets.properties*, lists the portlets affected by the phase provider for a specific partition. For development or testing purposes, you can override the phase provider for individual portlets.

The structure of the portlets properties file is simple. Each portlet resides on a separate line, followed by the = sign, as shown in this example:

```

wm_task_chart__taskchart=
wm_task_search__taskinboxsearchbar=
wm_task_search__taskinboxsearchresults=

```

```
wm_task_search___tlmsearchresults=
wm_task_search___tlmsearchquery=
```

A portlets properties file is not required for a partition in which all portlets are enabled. In this case, you can omit the `initFile` and `initFileComponentsEnabled` attributes in the phase provider file, which would otherwise manage use of the portlets properties file.

Temporarily Overriding the Phase Provider File

For development or testing, you can temporarily override the default behavior of the phase provider for an individual file by prefixing a + or - sign to the = sign, as follows:

- +=: Enables the portlet, regardless of the phase provider.
- -=: Disables the portlet, regardless of the phase provider.

Say, for example, the `initFileComponentsEnabled` attribute is `false`, which causes all portlets listed in the portlets properties file to be disabled. To enable any portlet in the list above, change = to +=, such as this:

```
wm_task_search___processearchbar+=
```

When you later remove the + sign, the portlet will be disabled along with the other portlets in the list.

The Default Partition Configuration

By default, there is one partition, named `default`, that includes all cluster nodes. The set of configuration files for the default partition is:

- `clusterPartitions.xml`—In the default state, this file defines only the default partition.
- `phaseProvider.xml`—This phase provider file supports the default partition. These attributes are set:

```
initFile="config:/defaultPartitionPortlets.properties"
initFileComponentsEnabled="false"/>
```

- `defaultPartitionPortlets.properties`—This file is empty by default.

In the default configuration, all portlets in the default partition are enabled.

Example: Creating Cluster Partitions

The following procedure shows the creation of cluster partitions by editing configuration files. This procedure is intended as a demonstration and is not a real-world example. Prior to this procedure, a cluster exists, having five nodes:

```
default
node2
node3
```

node4

node5

The example creates three partitions for the cluster:

default (all portlets are enabled)

partition2 (a specified group of portlets is disabled)

partition3 (a specified group of portlets is disabled, but some are temporarily enabled)

➤ **To create partitions through editing configuration files**

1. At a command line prompt, type the following command to move to the server's bin directory:

```
cd Software AG_directory\MWS\bin
```

2. To retrieve the clusterPartitions.xml file from the My webMethods Server database, type this command:

```
mws getconfig clusterPartitions.xml
```

3. To retrieve the phaseProvider.xml file from the My webMethods Server database, type this command:

```
mws getconfig phaseProvider.xml
```

4. To retrieve the defaultPartitionPortlets.properties file from the My webMethods Server database, type this command:

```
mws getconfig defaultPartitionPortlets.properties
```

5. Open the downloaded clusterPartitions.xml file in a text editor.

You can find all the downloaded files at this location:

Software AG_directory \MWS\server\serverName\config

6. Edit the clusterPartitions.xml file using the syntax in [“The Cluster Partition File” on page 229](#) to add partition2 and partition3 to the cluster.

The modified clusterPartitions.xml file looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<ClusterPartitions>
  <Partition name="default" frontEndUrl="http://my.company.server:8585">
    <Component name="default">
      </Component>
    </Partition>
  <Partition name="partition2">
    <Component name="node2">
      </Component>
    <Component name="node3">
```

```

    </Component>
  </Partition>
  <Partition name="partition3">
    </Partition>
  </ClusterPartitions>

```

Based on this file:

- The default partition contains the default node
- partition2 contains node2 and node3
- partition3 contains node4 and node5 (the last partition in the file contains all nodes not specifically assigned to other partitions)

7. Save and close the clusterPartitions.xml file.

As a result, the default partition is established according to the “The Default Partition Configuration” on page 231, with all portlets enabled.

8. Make a copy of phaseProvider.xml and name it partition2PhaseProvider.xml.

9. Open the partition2PhaseProvider.xml file in a text editor.

10. In the portlet phase nested inside the CoreServices phase, change defaultPartitionPortlets.properties to partition2Portlets.properties so the file looks like this:

```

<PhaseInfo name="portlet" enabled="true"
  class="com.webmethods.portal.service.portlet.impl.PortletProvider"
  initFile="config:/partition2Portlets.properties"
  initFileComponentsEnabled="false"/>

```

The phase provider now points to partition2Portlets.properties. Because initFileComponentsEnabled is set to false, portlets listed in that file will be disabled.

11. Save and close the partition2Portlets.properties file.

12. Make a copy of defaultPartitionPortlets.properties and name it partition2Portlets.properties.

13. Open the partition2Portlets.properties file in a text editor and add a list of the portlets that are to be disabled in partition2.

For example:

```

# These portlets will not be initialized.
wm_task_chart__taskchart=
wm_task_search__taskinboxsearchbar=
wm_task_search__taskinboxsearchresults=
wm_task_search__tlmsearchresults=
wm_task_search__tlmsearchquery=

```

In the user interface, these portlets have the following uses:

- taskchart displays Task Charts
- tlmsearchquery displays Task List Management
- The remaining portlets display My Inbox

14. Save and close the partition2Portlets.properties file.

As a result, certain portlets will be disabled on partition2 (node2 and node3).

Note:

The procedure steps for partition3 demonstrate how to temporarily override a phase provider file. As a result, the steps are abbreviated because they largely duplicate the steps for partition2.

15. Make a copy of phaseProvider.xml and name it partition3PhaseProvider.xml.

16. In partition3PhaseProvider.xml change, defaultPartitionPortlets.properties to partitionPortlets.properties.

17. Make a copy of partition2Portlets.properties and name it partition3Portlets.properties.

As a result, partition3 is configured to disable the same portlets as partition2.

18. In partition3Portlets.properties, prefix the = at the end of the last two portlets with a +.

For example:

```
# These portlets will not be initialized.  
wm_task_chart__taskchart=  
wm_task_search__taskinboxsearchbar=  
wm_task_search__taskinboxsearchresults=  
wm_task_search__tlmsearchresults+=  
wm_task_search__tlmsearchquery+=
```

Note:

The choice of portlets is arbitrary for this example.

As a result, partition3 (node4 and node5) has the same portlets disabled as partition2, but a subset of the portlets are enabled.

19. To deploy the new and revised files to the My webMethods Server database, type this command for each file:

```
mws putconfig fileName
```

20. Delete all of these files from the \serverName\config directory.

If you do not delete the files, cluster will continue to use the local version of the configuration file.

21. Restart each node in the cluster using this command:

```
mws - serverName restart
```

Changes to configuration files are not applied until after a restart.

Verifying That the Components Are Disabled

To verify that portlets are disabled in the scenario described in [“Example: Creating Cluster Partitions” on page 231](#), you would take the following general actions.

1. In the clusterPartitions.xml file, assign a separate `frontEndUrl` attribute for each partition.
2. Browse to the Front End URL of the default partition2 and log in as My webMethods Administrator.
3. In the navigation pane, click Task List Management.

This choice corresponds to these portlets:

```
wm_task_search___tlmsearchresults=
wm_task_search___tlmsearchquery=
```

The Task List Management displays without errors

4. Browse to the Front End URL of partition2 and log in as My webMethods Administrator.
5. In the navigation pane, click Task List Management.

Because the portlets are disabled, you will see an error.

Viewing Partitions in the My webMethods Server User Interface

To create and modify partitions, you need to use the configuration files described in [“Creating and Modifying Partitions” on page 228](#). But you can view information about existing partitions in the My webMethods Server User Interface.

➤ To view information about partitions

1. To navigate to the correct page, do one of the following:
 - In My webMethods: **Navigate > Applications > Administration > My webMethods > Cluster Settings > Cluster Partitions.**
 - As system administrator: **Administration Dashboard > Configuration > Cluster Administration > Cluster Partitions.**

The following table lists the information that the **Cluster Partitions** page displays:

Column Name	Purpose
Name	The partition name as defined in the cluster partition file. You can click the partition name to display a page with information about the partition.
Front End URL	The Front End URL used by the partition.
Nodes	The number of nodes in the partition.
Partition Rule	The rule applied to the partition by the phase provider file, which determines whether components are Enabled or Disabled .
Managed Components	The number of components (web applications and portlets) in the partition that are managed by the portlets properties file. The status of managed components can differ from the Partition Rule set by the phase provider file. For status of individual components, see “The Components Tab of the Partition Page” on page 236.
Detail	 — A link to the page containing specific information about the partition.

- To find information about a specific partition, click the **Name** column or .

The Nodes Tab of the Partition Page

The **Nodes** tab of a partition page contains information about the nodes that are part of that partition. The information on this tab reflects the nodes defined in the cluster partition file.

The following table lists the information that My webMethods Server displays on the **Nodes** tab:

Column	Purpose
Name	The name assigned to the node, as defined in the cluster partition file.
Host	The name of the host machine on which the node resides.
HTTP Port	The HTTP port used by the node.
HTTPS Port	The port used by the HTTPS listener of the node. A value of 0 indicates the HTTPS listener is disabled.

The Components Tab of the Partition Page

The **Components** tab of a partition page contains information about the web applications and portlets that are part of that partition. The information on this tab reflects the settings you have created in the phase provider and partition properties files for the partition.

There are two tree views, depending on how you want to list the components:

- Categories - lists components by category.

- Portlets - lists components in order, regardless of category.

To switch between the two views, choose the **Category Tree View** option or the **Portlet Tree View** option.

The following table describes the details that My webMethods Server displays for the available components:

Column	Purpose
Component	A tree view of components. Expand a node to reveal the portlets that are part of the web application.
Category	(Portlets tree view only) The name of the category to which a component belongs. Examples are Administration and Communication.
Name	The name of the component as it is installed in My webMethods Server. This name is different from the descriptive name used in the Component column.
Status	The status of the component. The status icons are: <ul style="list-style-type: none">  The component is enabled in this partition.  The component is disabled in this partition.

III System Administrator Functions

14	Attribute Providers	241
15	Managing Security	251
16	Analysis, Reporting, and Troubleshooting	305
17	My webMethods Server Configuration	323
18	Managing My webMethods Server Content	339
19	Managing the User Interface	347
20	Working with the Common Directory Services API	369
21	Sending Mobile Notifications from My webMethods Server	375

14 Attribute Providers

■ What are Attribute Providers?	242
■ Using Attribute Providers	243
■ Managing the Display of Principal Attribute Providers	247

What are Attribute Providers?

There are a variety of ways to provide and use attributes belonging to My webMethods Server users. Basic to each user are the My webMethods Server attributes: home page, skin, and number of lines displayed per page. In addition, there are principal attribute providers:

The following table lists the available attribute providers and their functions.

Attribute provider	Description
Core Attributes	A set of core attributes such as user ID and email address. If the user is in the system directory service, some fields are editable. If the user is in an external directory service, all fields are read only. See “The Core Attributes Attribute Provider” on page 243.
User Profile	A rich set of user attributes that you can maintain regardless of which directory service the user is a member of. The attributes are stored in the My webMethods Server database. Once established for each user, the User Profile Attributes can be used for wiring globally within My webMethods Server. See “The User Profile Attribute Provider” on page 244.
LDAP	A set of attributes from the external directory service. You can specify which attributes are exposed from a given directory service. See “The LDAP Attribute Provider” on page 244.
Database	A set of attributes from an external database directory service. You can specify which attributes are exposed from a given directory service. See “The Database Attribute Provider” on page 245.
Notification	A set of addresses, such as email, at which the user can receive notifications from My webMethods Server. See “The Notification Attribute Provider” on page 247.
Dynamic	A set of attributes whose values can change depending on the roles the user is a member of. See “The Dynamic Attribute Provider” on page 247.

Principal Attribute Providers are useful because any attribute they expose can be made available as wiring for a portlet. For example, suppose a portlet uses a postal code to display certain information when a user views a page. If the postal code is provided by wiring from a Principal Attribute Provider, when the postal code attribute is modified within a directory service, the portlet uses the modified attribute value.

Principal Attribute Providers are not enabled by default. To enable them, use the Principal Profile Administration page, described in [“Managing the Display of Principal Attribute Providers” on page 247.](#)

Using Attribute Providers

The Profile page for a user, group, or role displays the various sets of attributes as well as memberships in groups or roles. Depending on the directory service to which a user or group belongs, you can edit some attributes, or expose them for use in global wiring.

To see an example of the Profile page, as a system administrator, in the standard links, click **My Profile**. The page displayed is the Profile page for SysAdmin.

To see the Profile page for any user, group, or role, follow this search procedure:

➤ **To find the Profile page for a user, group, or role**

1. As system administrator: **Administration Dashboard > User Management > Manage *User_type* > Search**.

where *User_type* is **Users, Groups, or Roles**.

2. Search for the user you want to edit. For more information, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).
3. In the search results, click any link in the row of the user, group, or role, or click .

The Profile page for the user, group, or role is displayed.

The Core Attributes Attribute Provider

The Core Attributes Attribute Provider contains a set of attributes such as user ID and email address. Some fields are editable, depending on directory service membership.

User Information Panel

For individual users, the contents of the Core Attributes Attribute Provider appears on the Profile page as the **User Information** panel. For users in the system directory service, some fields, such as the email address, are editable. To edit information in this panel, see [“Editing Information for a User” on page 152](#).

For members of external directory services, this information is not editable, but it is available for the global wiring feature described in [“Wiring a Principal Attribute to a Portlet Property” on page 398](#).

Group Information Panel

For groups, the contents of the Core Attributes Attribute Provider appears on the Profile page as the **Group Information** panel. For groups in the system directory service, some fields, such as the email address, are editable. To edit information in this panel, see [“Editing Group Information” on page 163](#).

For groups in external directory services, this information is not editable but it is available for the global wiring feature described in [“Wiring a Principal Attribute to a Portlet Property” on page 398](#).

Role Information Panel

For roles, the contents of the Core Attributes Attribute Provider appears on the Profile page as the **Role Information** panel. The fields are not editable, but the information is available for the global wiring feature described in [“Wiring a Principal Attribute to a Portlet Property” on page 398](#).

The User Preferences Attribute Provider

User preferences are basic to any user of My webMethods Server. A My webMethods Server Administrator, system administrator, or the individual user (if given permission) can edit these basic attributes. To edit user preferences, see [“Editing Information for a User” on page 152](#).

The User Profile Attribute Provider

The User Profile Attribute Provider has a rich set of user attributes that you can maintain regardless of which directory service the user is a member of. The attributes are stored in the My webMethods Server database. Once established for each user, the User Profile Attributes can be used with the global wiring feature described in [“Wiring a Principal Attribute to a Portlet Property” on page 398](#).

A My webMethods Server Administrator, system administrator, or the individual user (if given permission) can edit the User Profile attributes. You need to enter the user profile attributes individually for each user, or allow the user to do so. You can use all of the attributes or a subset that is appropriate to your needs. To edit the User Profile attributes, see [“Editing Information for a User” on page 152](#).

The LDAP Attribute Provider

The LDAP Attribute Provider displays a specified set of attributes from the external directory service to which a user or group belongs. The attributes displayed in the **LDAP Attributes** panel are not editable but they are available for the global wiring feature described in [“Wiring a Principal Attribute to a Portlet Property” on page 398](#).

Note:

If the LDAP Attribute Provider is not enabled by default. See [“Managing the Display of Principal Attribute Providers” on page 247](#).

This attribute provider is not applicable to users or groups in the system directory service. Similar attributes are included in the User Profile Attribute Provider described in [“The User Profile Attribute Provider” on page 244](#).

Displaying the LDAP Attribute Provider

You cannot modify the contents of the **LDAP Attributes** panel on a Profile page, but you can display it if needed. To search for the Profile page of a user or group, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).

Exposing LDAP Attributes from an External Directory Service

The LDAP Attributes Provider displays user attributes that are exposed from an external directory service. You can expose selected attributes that are then available for the global wiring feature.

➤ To expose LDAP attributes from an external directory service

1. As system administrator: **Folders > System > Service* > Directory > Principal Attribute Providers > *Provider_type***.

where *Provider_type* is **User Principal Attribute Providers** or **Group Principal Attribute Providers**.

* You might have to click **Next** multiple times to find **Service** displayed in the table.

2. For **Ldap Attributes**, click , and then click **Properties**.

You should now be on the Properties of LDAP Attributes page.

3. Under **LDAP Attribute Names**, click **Add**.

4. Type the attribute name exactly as it is used in the external directory service and click **OK**.

For example, the attribute name for email on a particular directory service might be `mail`.

5. Under **LDAP Attribute Titles**, click **Add**.

6. Type a display name for the attribute for use within My webMethods Server and click **OK**.

For example, for the `mail` attribute, you might type a display name of `Email`.

7. If you have multiple LDAP attributes, make sure the order in the **LDAP Attribute Names** and **LDAP Attribute Titles** lists are the same.

The order in which attributes and titles appear in the lists determine the order in which they are displayed in the **Ldap Attributes** panel on the Profile page.

8. Click **Apply**.

The Database Attribute Provider

The Database Attribute Provider displays a specified set of attributes from the external database directory service to which a user or group belongs. The attributes displayed in the **Database Attributes** panel are not editable but they are available for the global wiring feature described in [“Wiring a Principal Attribute to a Portlet Property” on page 398](#).

Note:

If the Database Attribute Provider is not enabled by default. See [“Managing the Display of Principal Attribute Providers” on page 247](#).

This attribute provider is not applicable to users or groups in the system directory service. Similar attributes are included in the User Profile Attribute Provider described in [“The User Profile Attribute Provider” on page 244](#).

Displaying the Database Attribute Provider

You cannot modify the contents of the **Database Attributes** panel on a Profile page, but you can display it if needed. To search for the Profile page of a user or group, see [“Searching for Existing Users, Groups, or Roles” on page 142](#).

Exposing Database Attributes from an External Directory Service

The Database Attributes Provider displays user or group attributes that are exposed from an external database directory service. You can expose selected attributes that are then available for the global wiring feature.

➤ To expose database attributes from an external database directory service

1. As system administrator: **Folders > System > Service* > Directory > Principal Attribute Providers > *Provider_type***.

where *Provider_type* is **User Principal Attribute Providers** or **Group Principal Attribute Providers**.

* You might have to click **Next** multiple times to find **Service** displayed in the table.

2. For **Database Attributes**, click , and then click **Properties**.

You should now be on the Properties of Database Attributes page.

3. Under **Attribute Names**, click **Add**.
4. Type the attribute name exactly as it is used in the external database directory service and click **OK**.

Important:

An attribute used here must be returned by the Query Lookup User by ID attribute in the database directory service. See [“Configuring an External Database Directory Service” on page 122](#).

For example, the attribute name for postal code on a particular directory service might be `zipcode`.

5. Under **Attribute Titles**, click **Add**.

6. Type a display name for the attribute for use within My webMethods Server and click **OK**.

For example, for the `zipcode` attribute, you might type a display name of `Zip Code`.

7. If you have multiple database attributes, make sure the order in the **Attribute Names** and **Attribute Titles** lists are the same.

The order in which attributes and titles appear in the lists determine the order in which they are displayed in the **Database Attributes** panel on the Profile page.

8. Click **Apply**.

The Notification Attribute Provider

The Notification Attribute Provider allows you to specify the various addresses at which a user can receive notifications. A My webMethods Server Administrator, system administrator, or the individual user (if given permission) can edit these basic attributes. To edit user preferences, see [“Editing Information for a User” on page 152](#).

Note:

The Notification Attribute Provider is not enabled by default. See [“Managing the Display of Principal Attribute Providers” on page 247](#).

User preferences are basic to any user of My webMethods Server. A My webMethods Server Administrator, system administrator, or the individual user (if given permission) can edit these. To edit user preferences and notifications, see [“Editing Information for a User” on page 152](#).

The Dynamic Attribute Provider

The Dynamic Attribute Provider allows you to provide an attribute for a role. A user, group, or role that is a member of a role has all dynamic attributes of the role. If a user is a member of multiple roles, and multiple roles have attributes with the same key, you can determine which role will have precedence. In addition, you can assign an attribute value to a user, overriding the attribute values provided by roles. Dynamic attributes are available for the global wiring feature described in [To edit user preferences, see “Editing Information for a User” on page 152](#).

Note:

The Dynamic Attribute Provider is valid only for roles. Users and groups have dynamic attributes based on roles of which they are members.

For information on functions you can perform on dynamic attributes, see [“Defining Dynamic Attributes Associated with a Role” on page 194](#).

Managing the Display of Principal Attribute Providers

If all principal attribute providers were displayed on a profile page by default, the page would be crowded and potentially difficult to read. On the **Principal Profile Administration** page, you can

choose which principal attribute providers to display on a profile page and the order in which they appear. You can set principal attribute providers for users, groups, and roles.

The following table lists the available principal attribute provider lists and the profile pages that each list applies to:

List	Profile page
USER Attribute Providers	The Profile page for a user. By default, the Core Attributes Provider is displayed as the User Information panel. Other default attribute providers are Groups, Roles, and User Preferences. You can add any Principal Attribute Provider applicable to an individual user.
GROUP Attribute Providers	The Profile page for a group. By default, the Core Attributes Provider is displayed as the Group Information panel. The default attribute providers are Groups and Group Members. You can add any Principal Attribute Provider applicable to a group.
ROLE Attribute Providers	The Profile page for a role. By default, the Core Attributes Provider is displayed as the Role Information panel. The other default attribute provider is Dynamic Attributes.

You can perform the following actions on the **Principal Profile Administration** page:

- Add a principal attribute provider to the profile page.
- Rearrange the position of principal attribute providers on the profile page.
- Remove a principal attribute provider from the profile page.

Adding a Principal Attribute Provider

To add a Principal Attribute Provider to a Profile page, use the following procedure.

➤ To add a Principal Attribute Provider to a Profile page

1. As system administrator: **Administration Dashboard > User Management > Principal Profile Administration**.
2. In the **USER, GROUP, or ROLE Attribute Providers** area, click **Add**.
3. To specify the Principal Attribute Provider want to add, move it to the **Selected Items** box and click **Select**.
4. At the bottom of the page, click **Apply**.

Changing the Display Order for Principal Attribute Providers

To change the order in which Principal Attribute Providers appear on a Profile page, use the following procedure.

➤ To change the display order for Principal Attribute Providers

1. As system administrator: **Administration Dashboard > User Management > Principal Profile Administration**.
2. In the **USER, GROUP, or ROLE Attribute Providers** area, To reorder Principal Attribute Providers, move them up or down as needed.

The first attribute provider in the list has the left-most position on the Profile page, followed by the second attribute provider, and so on.

3. At the bottom of the page, click **Apply**.

Removing a Principal Attribute Provider

To remove a Principal Attribute Provider from a Profile page, use the following procedure.

➤ To remove a Principal Attribute Provider from a Profile page

1. As system administrator: **Administration Dashboard > User Management > Principal Profile Administration**.
2. In the **USER, GROUP, or ROLE Attribute Providers** area, select a Principal Attribute Provider from the list and then click **Remove**.
3. At the bottom of the page, click **Apply**.

15 Managing Security

■ About My webMethods Server Security	252
■ Managing Authentication	263
■ Configuring Kerberos Authentication	266
■ Configuring NTLM Authentication	271
■ Configuring NTLMv2 Authentication	272
■ Configuring External Configuration Credentials	274
■ Configuring My webMethods Server Single Sign-On	277
■ Configuring OAuth 2.0 Authentication	285
■ Clearing Session Passwords from Memory	289
■ Retaining Session Passwords in Memory	289
■ Turning On or Off Auto Complete for Usernames and Passwords	290
■ Controlling the Number of Failed Login Attempts	290
■ Controlling Login IP Ranges	291
■ Encrypting Passwords for Global Environment Variables	293
■ Allowing Context Impersonation	294
■ Using Password Complexity Policies	294
■ Working with Response Header Rules	299

About My webMethods Server Security

My webMethods Server has many functions that contribute to its overall security infrastructure. When discussing security, it is always necessary to separate the discussion of authentication (Auth) from Authorization (AZ). While they are almost always related, the two concepts are distinct and work together to contribute to an overall security solution.

Authentication is defined as an assurance that a party to some computerized transaction is not an impostor. Authentication typically involves using a password, certificate, PIN, or other information that can be used to validate identity. The goal of authentication is to simply verify that “you are who you say you are.”

Authorization is defined as the process of determining, by evaluating applicable access control information, whether a party is allowed to have the specified types of access to a particular resource. Usually, authorization is in the context of authentication. Once a party is authenticated, that party may be authorized to perform different types of activities.

My webMethods Server provides built-in infrastructure for both authentication and authorization. My webMethods Server is also designed in a way that allows it to be extended so that existing security infrastructure can be re-used and leveraged for both authentication and authorization.

In My webMethods Server, you can apply both authentication and authorization to the entire server or to individual server resources, which include folders, pages, portlets, links, documents, files, or custom objects.

Server Authentication

My webMethods Server supports many different ways for users to identify themselves. These different methods are called Authentication Schemes. These schemes are simply different ways to gather user credentials and validate their authenticity.

Forms Authentication

Forms authentication is the default authentication scheme for My webMethods Server. This authentication scheme presents a form to a user and gathers the necessary credentials that are passed to the server by means of a form POST (data is passed to the server’s standard input). It is simple to customize the form or page that is used for authentication. It is also easy to present different forms and pages based on a wide variety of different criteria. For example, it is very likely that you would want to provide different login experiences for users accessing a server from mobile devices than for users accessing the server from a browser.

Anonymous Authentication

The anonymous authentication scheme is used when you do not want to challenge users for credentials. My webMethods Server honors an anonymous request and establishes a session with the server as a special Guest user, but the user is never prompted for credentials. The anonymous authentication scheme is used for unprotected areas of the server that might be public facing and do not contain sensitive information. By associating the request with a session and a user ID of

Guest, an administrator can extend behaviors of anonymous access by controlling permissions of the Guest user. Guest is one of the default users installed as part of the system directory service. It is also possible to track session activity of anonymous users for reporting requirements.

Basic Authentication

Basic authentication is one of the original and most compatible authentication schemes for programs using HTTP as a transport mechanism. Unfortunately, it is also one of the least secure, as it sends the username and password unencrypted to the server. The credentials are typically passed in as HTTP header parameters. The user experience for basic authentication is a popup window that renders in the native windowing system. For example, when you use basic authentication on Windows, a Windows dialog box opens to prompt the user for credentials before the request can be honored.

Kerberos Authentication

Kerberos is a network authentication protocol that allows secure communication between clients and My webMethods Server over a non-secure network. The Kerberos authentication scheme provides mutual authentication for clients and My webMethods Server, and validates identity using symmetric encryption and a trusted third-party key distribution center (KDC).

You can configure My webMethods Server to use Kerberos authentication integrated with Windows authentication for single sign-on. With this authentication scheme, a valid My webMethods Server user, already authenticated by Windows can access My webMethods Server without providing additional credentials.

To configure single sign-on using Kerberos and Windows authentication, see [“Configuring Kerberos Authentication” on page 266](#).

NTLM Authentication

NTLM (Windows NT LAN Manager) is an authentication protocol used in various Microsoft network protocol implementations and supported by the NTLM Security Support Provider (NTLMSSP). Originally used for authentication and negotiation of secure DCE/RPC, NTLM is also used throughout Microsoft's systems as an integrated single sign-on mechanism. On Windows deployments, when NTLM is set up and configured as an authentication scheme for My webMethods Server, users do not need to re-authenticate for server resources if they are already logged into a Windows domain. You can choose either NTLM or NTLM version 2 (NTLMv2) authentication.

To use NTLM authentication you need to explicitly specify the Primary Domain Controller for the domain, as described in [“Configuring NTLM Authentication” on page 271](#).

To use NTLMv2, you need to purchase the Jespa Java software library from IOPLEX Software. For information on configuring NTLMv2, see [“Configuring NTLMv2 Authentication” on page 272](#).

Note:

You can use either the webMethods NTLM authentication or NTLMv2, but not both at the same time.

HTTP Header Authentication

My webMethods Server can be configured to accept External HTTP authentication credentials from third-party security and access control products (such as Computer Associates, Oblix, and so forth). These credentials are case sensitive, depending on platform and web server and are most likely to be headers such as `sm_user` or `SM_USER`.

When you configure and set up HTTP header authentication within My webMethods Server, the server uses credentials from a third-party authentication engine. Typically, these third parties use a security agent to intercept the request prior its getting to the server. The basic flow of events in this request is:

1. The user attempts to go to a server resource.
2. Prior to connecting to the server, if the third-party security agent does not see the proper credentials, the agent redirects the user to a mechanism that gathers credentials.
3. The user provides the credentials and is then redirected back to the server resource.
4. The server reads the appropriate HTTP header and maps the user appropriately.

To configure this interaction between the server and the third-party security agent, you need to take these actions.

5. After My webMethods Server installation, configure the third-party product to protect the server, which typically involves creating a policy that protects the server URL.
6. Verify that the server and the third-party security product are configured to look at the same directory store. For more information on directory services, see [“Managing External Data Sources” on page 129](#).
7. Configure the server to look for the right HTTP header. For more information, see [“Configuring External Configuration Credentials” on page 274](#).

Note:

In the case of SiteMinder from Computer Associates, it is also necessary to specify the Logout URI in SiteMinder. In the SiteMinder Administrator applet, modify the `logoutURI` attribute to be `'/?method=logout'` (without the quotes)

Important:

The HTTP Header Authentication Administration page should only be enabled if you are using a third-party security provider. After the page is enabled, the server acts as though all users have been authenticated.

Extended and Extensible Authentication Schemes

My webMethods Server provides hooks for developers to provide their own custom authentication schemes. To develop a custom authentication scheme, create a portlet, implement the correct

interfaces, and register it with the server. Once created, the new authentication scheme participates in the security infrastructure just like any other authentication scheme that is provided as part of My webMethods Server.

My webMethods Server has a concept of a default authentication scheme that is applied to an entire deployment. A newly configured server uses forms as its default authentication scheme. The server challenges initial requests for protected resources with a form requiring the user to type a user name and password.

At any time, you can change the default authentication scheme for a server to one of the registered authentication schemes. For more information, see [“Specifying a Default Authentication Scheme” on page 264](#).

Every server resource can have an authentication scheme that overrides the setting for the entire deployment. For example, you might have one set of pages and portlets that are completely anonymous and others that require user credentials to be presented. You would do this by associating the anonymous authentication scheme with the resources that do not require authentication. For information on managing authentication schemes for individual server resources, see [“Assigning an Authentication Scheme to a Server Resource” on page 264](#).

Extending Login and Splash Page Behavior

To understand the login process and flow of events, it helps to analyze an example of how a system administrator would extend a deployment to have custom login page behavior. The following set of steps uses the concepts of anonymous access, forms-based authentication, and login pages to form a solution. Some of the steps require developer knowledge.

1. Design a page that has a login portlet on it. Once the page is created, set the authentication scheme of the page to “anonymous” so everyone can get to the page and be presented with the login portlet.

Optionally, you can set access rights on other parts of the page so that the login page has different appearances, depending on the identities of users. To address even broader requirements of personalizing the login page, it is also easy to set up custom login pages based on rules themselves.

2. After setting the authentication scheme of the page to anonymous, make sure the login portlet itself can be seen by a Guest user. For more information about the Guest user, see [“Managing Directory Services” on page 110](#).

You may also want to modify the look and feel of the page by removing title bars, adding explicit instructions, or implementing other business requirements.

3. You can control where a user is redirected after login. In the Properties page for the login portlet, modify the Login Target property to the page where the user is redirected. Keep in mind that the Login Target be static or it can be an alias. If you use an alias like `/user.current.start.page`, you can alternatively set up start page rules to govern different start pages based on information about the user logging in.

It is also possible to redirect a request, if not authenticated, to go to the appropriate login page. To do so, modify the Redirect URI property of the authentication scheme assigned to the page. When an unauthenticated user requests the page, the user is redirected to the specified page. As with login targets, a redirect URI can be either static or an alias.

Making a Custom Authenticator Available to Integration Server

If you implement a custom authenticator portlet to be used with a database directory service for My webMethods Server, and if Integration Server is configured to use Common Directory Services, you need to take additional steps to enable authentication for users logging into Integration Server.

➤ To make a custom authenticator available to Integration Server

1. Stop all webMethods components.
2. Open the custom authenticator .pdp file using the .zip extractor of your choice.
3. Extract the `/WEB-INF/lib/custom_authenticator.jar` from the .pdp (the name will vary with the naming convention of your custom component).
4. To make the custom authenticator available to all Integration Server instances, copy the extracted .jar file into this location:

Integration Server_directory /lib/jars/

5. To make the custom authenticator available to a specific Integration Server instance, copy the extracted .jar file into this location:

Integration Server_directory /instances/serverName/lib/jars/

6. Modify Common Directory Services configuration files by doing the following:

- a. Locate the following .jar file:

Software AG_directory \common\lib\wm-mws-library.jar

- b. Extract these two files from the `wm-mws-library.jar` file.

`initPortlets.properties`
`initXTypes.properties`

- c. Open each of these files in a text editor and use this syntax to add the name of the custom authenticator portlet as the last line of the file:

`custom_authenticator=`

Use the other portlet names in these files as examples.

- d. Save each file and repackage them both in the `wm-mws-library.jar` file.

7. Restart all webMethods components.

Security Assertion Markup Language

My webMethods Server supports single sign-on through the Security Assertion Markup Language (SAML), an XML-based framework for the exchange of security information. Using SAML, an application on a target computer grants access based on an assertion from the source computer.

My webMethods Server can be the calling program, or Security Provider, or can be configured to authenticate the user sign-on for a target web application. For more information, see [“Configuring My webMethods Server Single Sign-On” on page 277](#).

My webMethods Server supports third-party identity provider (IDP) initiated single sign-on. You can configure My webMethods Server to use SAML 2.0 for exchanging authentication and authorization data between My webMethods Server (Service Provider) and a third-party IDP. In this case, My webMethods Server will be a SAML consumer and the third-party identity provider (IDP) will be the SAML authority. For information about configuring third-party IDP initiated single sign-on, see [“Using Single Sign-On with SAML and a Third-Party Identity Provider” on page 280](#).

OAuth 2.0 Authentication

You can configure an OAuth 2.0 authentication flow in My webMethods Server, and log in to My webMethods with credentials from an external identity provider, such as Google, Twitter, or Salesforce. In this authentication flow, My webMethods Server uses the OAuth 2.0 protocol with the OpenID Connect identity layer as an authentication and authorization method.

When logging in using the OAuth 2.0 authentication flow, My webMethods Server redirects you to the authorization endpoint of the external identity provider, for example the Google Authorization Server, where you supply your credentials. After authentication, the identity provider redirects you back to a My webMethods Server endpoint. My webMethods Server acquires ID and Access tokens from the provider, validates the ID token, and requests a list of UserInfo claims. Based on the claims returned by the provider, My webMethods Server registers an internal user account and associates the account with a login session.

To configure OAuth 2.0 and OpenID Connect authentication flow in My webMethods Server, see [“Configuring OAuth 2.0 Authentication” on page 285](#).

Server Authorization

After a user request has been authenticated by the server, it is usually necessary to do some authorization checks to make sure that the user making the request has the necessary privilege to act on that resource. In My webMethods Server, the most common way to do authorization checks is by evaluating Access Control Lists (ACLs). ACLs can be associated with every kind of server resource, such as pages, portlets, and so forth.

More advanced concepts like verbs and mechanics (groupings of business logic) are server resources as well, and therefore also participate in the ACL evaluation model. This feature allows developers to programmatically lock down capabilities of the server.

To understand the authorization engine in My webMethods Server, look at the composition of an ACL. An ACL is a list of Access Control Entries (ACEs). An ACE is a simple structure containing an element called a Principal and an element called a Right Set.

A *Principal* is a user, group or role. The following table lists examples of principals:

Principal	Example
User	Myles Perkins
Group	Members of the Perkins family
Role	A role definition that resolves to Myles, such as, "Users who have the 'Job Title' attribute value set to 'Product Manager.'"

Right Sets are groupings of actions that can be performed on a server resource. An example of a Right Set is "Grant the ability to read. Right Sets themselves are broken down into two distinct parts, Capabilities and Settings. Different types of server resources have different Capabilities associated with them. For example, pages have Capabilities that include "Add Portlet To Page" while folders have Capabilities that include "Create Sub Folder" and "Can Read Items in this Folder."

The other part of a Right Set, the Setting, can have four possible values: DELEGATE, DENY, GRANT or NONE. Each Capability that makes up a Right Set has a Setting value. Right Sets are made up of many Capability-Setting pairings. Here is an example of a Right Set:

DENY + create sub folder

GRANT + read

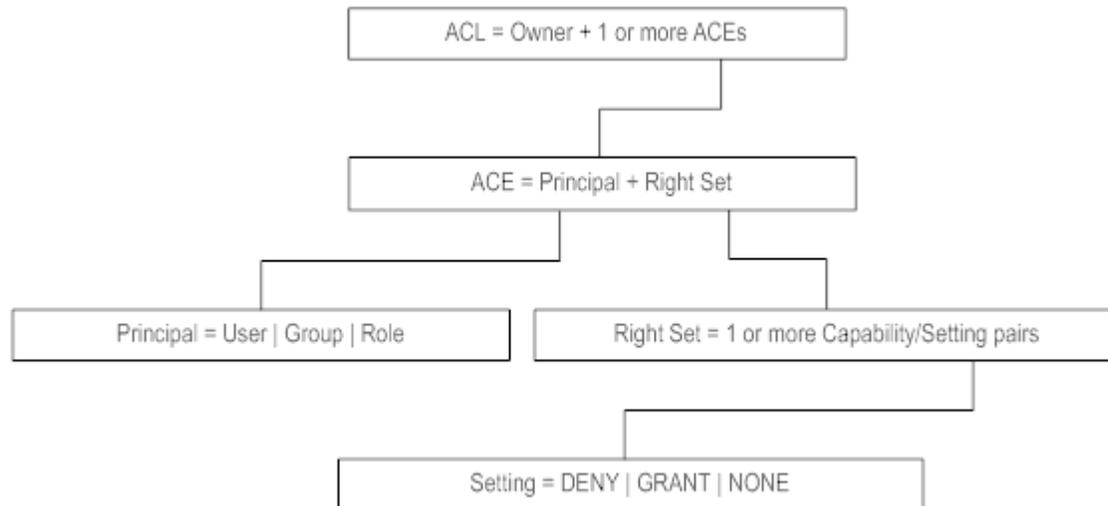
This particular Right Set is made up of two Capability-Setting pairings. If associated with a folder resource, this Right Set is resolved to deny a Principal the ability to create sub folders but grant the ability to actually read the folder.

The effects of each setting are described in the following table:

Setting	Effect
DENY	Denies the access to perform the capability.
GRANT	Explicitly grants access to perform the capability.
DELEGATE	Explicitly grants access and gives the right to assign the capability to another Principal.
NONE	Provides no explicit Setting. Authorization for this server resource will be determined from another source.

The following figure shows the relationships described in this section.

Anatomy of an ACL



As an example, to deny read access to Brian and the Marketing Group from the Engineering page, we would have the following setup:

- The server resource is the Engineering page.
- The Engineering page has an ACL associated with it that contains the ACEs, listed in the following table:

Principal	Right Set
Brian	DENY read
Marketing Group	DENY read

Controlling Permissions on Server Resources

If you have authorization to change access to a server resource, you use the Permissions portlet of My webMethods Server to assign access control to it. For example, if you are the owner of the Engineering folder in the example described in [“Server Authorization” on page 257](#), a wizard in the Permissions portlet allows you to select one or more Principals and Right Sets, and associate them with that folder. For more information on controlling permissions, see [“Managing Permissions” on page 169](#).

You do not have to explicitly grant and deny access for every newly created object. If you give your taxonomy a little forethought, you can keep the potential maintenance burden to a minimum. My webMethods Server employs a method called *static propagation* of its access rights on server objects when they are created. This means that at creation time, a server resource receives its access rights from its parent resource. If subsequent changes are made to the parent's access rights, these rights are not dynamically updated in the child object. However, you can use the Permissions portlet to cause parent objects to apply access rights explicitly to their children.

To illustrate static propagation and parent-child interaction as it relates to access rights, we will return to the example of the Engineering folder. In that example, the Engineering folder has BRIAN DENY. As the owner of the Engineering folder, a user creates a sub folder called Secret Project.

Because of static propagation, the new Secret Project folder has BRIAN DENY at the time of creation. If the owner goes back and changes the permissions of the Engineering Folder to allow Brian access, Brian still does not have access to the Secret Project sub folder.

Authorization Determination

Now that you have a background in the concepts of making an authorization decision, you can see how access is actually determined at run time. When a server resource is requested, the server evaluates the ACL associated with that resource against the context in which the current request is generated. If a user requests access to a page, the ACL for that page is evaluated to determine whether the user request should be honored.

There are a few simple rules in determining authorization that handle a large percentage of any conflicts that may arise:

- DENY always takes precedence over Allow (It is good to be paranoid in dealing with security)
- Users always take precedence over groups and roles

To illustrate these rules and how they are applied to resolve conflict, we return to the example Engineering folder. In the following example, there are three ACE entries in the ACL associated with the Engineering folder:

```
BRIAN + DENY READ
```

```
MARKETING GROUP + DENY READ
```

```
BRIAN + GRANT READ
```

If Brian is a member of the Marketing group (and even if he wasn't) he is denied access to the Engineering folder. The user-based ACE takes precedence over the group-based ACE so the MARKETING GROUP ACE has no effect. Subsequently, the conflict between BRIAN being granted and denied access is resolved by denying access because DENY always wins.

Lists, pages, child objects and Searches

As mentioned earlier, a Principal can be a user, group, or role. Information about a Principal comes from a directory service. My webMethods Server has an embedded system directory service, described in [“Managing Directory Services” on page 110](#), as well as the ability to tie to external directory servers. Examples of these external directory servers are Active Directory, LDAP servers, ADAM, and an RDBMS. In addition, group and role information for My webMethods Server authorization decisions is determined when a user logs into the server. If a user's group membership changes during an active session, the change is not reflected in the server until the user logs out and logs back in. For more information about users, groups, roles, and directory services, see [“Managing Users and Groups” on page 149](#).

Security Realms

My webMethods Server provides a feature called Security Realms to augment its security model. *Security Realms* are collections of server resources that share the same ACL. The use of Security Realms makes it possible to easily manage permissions on large numbers of server resources. By

adding the resources directly to a Security Realm, a system administrator can add Principal information to that realm to control access.

Security Realms become very useful if you have a large number of server resources and only a few access levels. For example, you may have a large customer-facing server that has a large number of portlets, pages and areas of taxonomy. However, this server may only have three levels of access that need to be managed: Gold, Silver and Bronze. With each level represented by a Security Realm with the appropriate pages, portlets and taxonomy elements in them, a system administrator needs only to add a new customer to the appropriate Security Realm, granting the customer the correct level of access. Likewise, changing a customer from one level to another is a simple one-step operation.

Used in the appropriate deployments, Security Realms add value, not only by minimizing the administrative burden, but by greatly reducing the number of underlying records required to support the security model. For example, assume a server has 500,000 server resources and you are managing permissions for 50 users, all of whom have the same access:

- Managing permissions by ACL requires 25 million records in the My webMethods Server database.
- Managing permissions by Security Realm uses one Security Realm and one role with 50 members, requiring a total of three records in the My webMethods Server database.

It should be noted that if a server resource is added to a Security Realm, the Security Realm access control has precedence over an individual ACL and authentication scheme for that resource.

For information on managing Security Realms, see [“Using Security Realms” on page 176](#).

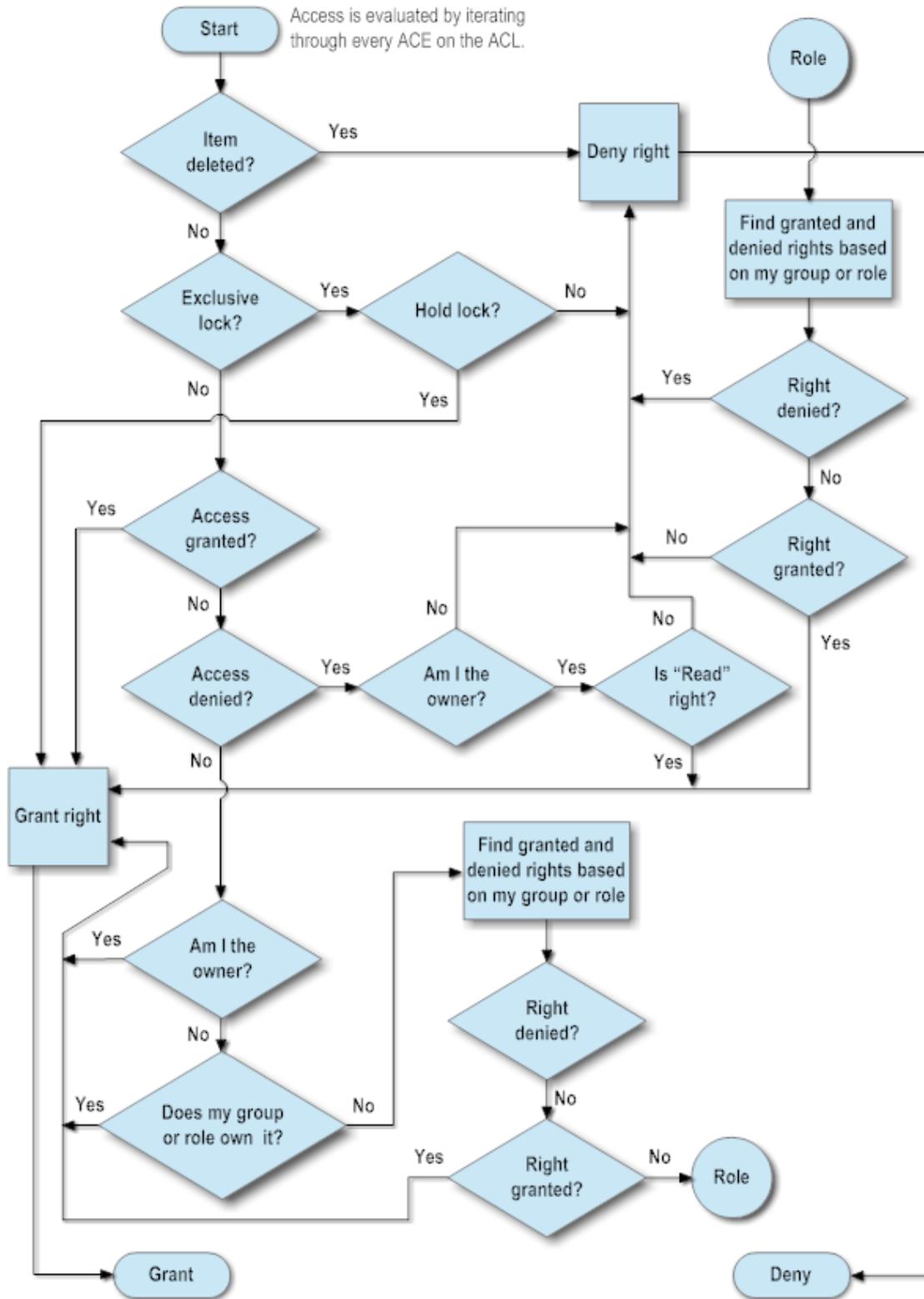
Server Verbs and Access Control

A *server verb* is an operation such as publishing, deleting, updating, subscribing, and setting permissions, which is available through the My webMethods Server API. As noted earlier, server verbs are server resources that can also participate in the security model of the server. In this way, one can control granular access to server capabilities programmatically as well as through the Administrative Dashboard. It should be noted that server verbs typically have two levels of security checks, performed in this order:

1. Does the user have access to the server verb itself?
2. Does the user have the rights to the resource upon which the server verb is trying to act?

A system administrator can control access to server verbs using the Security Realms Administrative page. My webMethods Server ships with default Security Realms to help administrators manage access to different server capabilities. For information about the default Security Realms, see [“Using Security Realms” on page 176](#).

Authorization Decisions in My webMethods Server



Managing Authentication

An authentication scheme is a way to gather user credentials and validate their authenticity. Within My webMethods Server, you can manage authentication for a server as a whole by specifying a default authentication scheme. As delivered, the forms authentication scheme is the default for all server resources. In addition, every server resource can have an authentication scheme that overrides the setting for the entire deployment.

Note:

A Security realm always takes precedence over an authentication scheme.

The following table lists the authentication schemes that My webMethods Server uses:

Scheme	Description
Anonymous	Allows unrestricted access to a server resource. Used for unprotected areas of the server that might be public facing and do not contain sensitive information. Because a user is not challenged for credentials, the anonymous authentication scheme is appropriate for login pages.
Forms	Presents a form to an unauthenticated user and gathers the necessary credentials that are passed to the server. The forms authentication scheme is the default for all server resources because it redirects unauthenticated requests to a default login page.
Basic	Typically passes credentials as HTTP header parameters. The user experience for basic authentication is a popup window that renders in the native windowing system.
Kerberos	Enables single sign-on for users on Windows. My webMethods Server users, already authenticated by Windows, need not login again to access My webMethods Server. For more information, see “Configuring Kerberos Authentication” on page 266 .
HTTP Header	Accepts external HTTP authentication credentials from third-party security and access control products (such as Computer Associates, Oblix, and so forth). After this authentication scheme is enabled, the server ignores all other authentication schemes. For more information, see “Configuring External Configuration Credentials” on page 274 .
NTLM	Used for authentication in various Microsoft network protocol implementations. On Windows deployments, when the NTLM authentication scheme is the default for a server, users do not need to re-authenticate for server resources if they are already logged into a Windows domain. For more information on NTLM authentication, see: <ul style="list-style-type: none"> ■ “Configuring NTLM Authentication” on page 271 ■ “Configuring NTLMv2 Authentication” on page 272

Scheme	Description
SAML	Supports single sign-on (SSO) through the Security Assertion Markup Language (SAML). Using SAML, an application on a target computer grants access based on an assertion from the source computer. For more information about SSO, see “Configuring My webMethods Server Single Sign-On” on page 277 .
OAuth 2.0	Allows users to login to My webMethods and access layered applications or other server resources, using credentials from a third-party identity provider. For more information, see “Configuring OAuth 2.0 Authentication” on page 285 .

Specifying a Default Authentication Scheme

When My webMethods Server is initialized, the forms authentication scheme is the default. The forms authentication scheme redirects unauthenticated requests to a default login page. You can change the default authentication scheme for My webMethods Server to one of the registered authentication schemes.

Note:

Do not use this procedure if you intend to use the httpHeader authentication scheme to accept credentials from third-party security providers. Instead, use the HTTP Header Authentication Administration page, described in [“Configuring External Configuration Credentials” on page 274](#).

➤ To change the default authorization scheme for My webMethods Server

1. As system administrator, click **Administration Dashboard > Configuration > Alias Management**.
2. On the **Keyword** tab, type `auth.scheme.default` and click **Search**.
3. Click  for `auth.scheme.default`.
4. On the Update Alias panel, click **Browse** and browse to **Folders > System > Authentication Schemes**.
5. Move the appropriate authentication scheme to **Selected Items** and click **Select**.
6. Click **Update**.

Assigning an Authentication Scheme to a Server Resource

Every server resource can have an authentication scheme that overrides the default authentication scheme for My webMethods Server. The specific procedure depends on whether or not the resource

is a member of a Security Realm. The valid authentication schemes are described in “Managing Authentication” on page 263.

➤ **To assign an authentication scheme for an individual server resource**

1. As system administrator: **Home > Folders** and then navigate to the individual server resource.
2. Click , and then click **Permissions**.
3. Click the **Security Realm** tab.
4. If the **Security Realm Name** field displays **No Security Realm Assigned**, do the following:
 - a. Click the **Authentication** tab.
 - b. From the **Authentication Scheme** list, choose the authentication scheme to apply to the server resource and click **Apply**.
5. If the **Security Realm Name** field displays a Security Realm name, do the following:
 - a. Copy the Security Realm name, paste it into the input field of the page banner, and click **Search**.

As an alternative, you can navigate to **Folders > Administrative Folders > Administration Dashboard > Configuration > Security Realms Administration** to locate the Security Realm.

- b. Click  and then click **Permissions**.
- c. Click the **Authentication** tab.
- d. From the **Authentication Scheme** list, choose the authentication scheme to apply to the Security Realm and click **Apply**.

Redirecting a User After Login

By default, when a user logs into a server, redirects the user to the same page. You can alter the Login Target property of the login portlet so a successful login redirects the user to a page of your choosing.

➤ **To redirect a user to another page after login**

1. At the right edge of the title bar for the login portlet, click  and then click **Properties**.
2. For the **Login Target** property, do one of the following:

- Click **Browse** and then move the target page to the **Selected Items** box and click **Select**.
- Click **Use Alias**, and then type the alias of the page to which the user should be redirected in the **Alias Name** field. Click **Test** to determine if the alias is valid and the alias target is the correct one. If the alias is correct, click **Select**.

Note:

If you type `/user.current.start.page`, the user is redirected to the start page defined by start-page rules.

3. Click **Apply**.

Redirecting an Unauthenticated Request

Using the forms authentication scheme, an unauthenticated request to a protected server resource results in the user being redirected to a default login page. Alternatively, you can redirect the user to a target of your choosing, whether it is a custom login page or a page that provides unprotected content.

➤ To redirect an unauthenticated request to a different page

1. As system administrator, click **Administration > Folders > System > Authentication Schemes**.
2. Click  for the default authentication scheme, and then click **Properties**.

By default, the forms authentication scheme redirects unauthorized requests to the default login page. You can verify that an authorization scheme is the default by looking at the Properties page. The **Aliases** list contains the `auth.scheme.default` alias.

3. In the **Performs Redirect** list of the Properties page for the authentication scheme, make sure **Yes, this performs a redirect** is selected.
4. Under **Redirect URI**, do one of the following:
 - Click **Browse**, and then move the target page to the **Selected Items** box and click **Select**.
 - Click **Use Alias**, and then type the name of the alias to which the user should be redirected in the **Alias Name** field. Click **Test** to determine if the alias is valid and the alias target is the correct one. If the alias is correct, click **Select**.

5. Click **Apply**.

Configuring Kerberos Authentication

Perform the following high-level steps to enable single sign-on for My webMethods Server with Windows Integrated Authentication and Kerberos:

1. Configure Active Directory, and the Windows Server that hosts Active Directory as the KDC for Kerberos authentication
2. Configure a directory service for the Active Directory, and Kerberos authentication in My webMethods Server
3. Configure web clients for single sign-on with Kerberos.

If authenticating a request using SSO with Kerberos fails, for example, because an exception occurs when processing the Kerberos token, My webMethods Server uses the default forms authentication scheme and redirects unauthorized requests to the login page. If you change the default authentication scheme or the target page for unauthorized requests, My webMethods Server uses the new default scheme, or page.

When your environment includes multiple web clients, you can configure only specific clients to use Kerberos authentication. Unconfigured web clients display a blank page when the Kerberos authentication scheme is used for logging in to My webMethods Server. In this case, users can refresh the web clients to use the default authentication scheme and log in using the login page.

Configuring Windows Server and Active Directory for Kerberos Authentication

Perform the following steps to configure Active Directory as the key distribution center (KDC) for Kerberos authentication. Unless indicated otherwise, perform the steps on the Windows Server machine that hosts Active Directory.

➤ To configure Active Directory as the key distribution center for Kerberos authentication

1. Configure a user account for the Kerberos principal in Active Directory. Do not select any encryption. The default encryption is RC4.
2. Create a Service Principal Name (SPN) and a keytab file using the ktpass command line utility and the following command:

```
ktpass -out <Keytab_File_Name>.keytab -princ
HTTP/<FQDN_of_Active_Directory_Server>@<Domain_Name> -mapUser
<FQDN_of_Active_Directory_Server>@<FQDN_of_MWS_Server_Machine> -mapop
set<MWS_Server_User_Password> -crypto all -ptype KRB5_NT_PRINCIPAL -kvno 0
```

Example:

```
ktpass -out MWS_Kerberos_User.keytab -princ HTTP/VMHOSTNAME.SPARTA.RNDLAB.
LOC@SPARTA.RNDLAB.LOC
-mapUser Bob@SPARTA.RNDLAB.LOC
-mapOp set -pass pass12345 -crypto all
-pType KRB5_NT_PRINCIPAL -kvno 0
```

Where `MWS_Kerberos_User` is the name of the keytab file, `Bob` is the user, and `SPARTA.RNDLAB.LOC` is the fully qualified domain name of the AD server.

3. Copy the keytab file to any directory of the machine that hosts My webMethods Server.

4. Verify that the keytab file is created correctly by executing the following java command from <JAVA_INSTALL>/jre/bin:

```
kinit -J-Dsun.security.krb5.debug=true -k  
-t <Keytab_file_absolute_path> HTTP/<FQDN_of_Active_Directory_Server>
```

Configuring My webMethods Server for Kerberos Authentication

Use the following procedure to configure My webMethods Server for Kerberos authentication.

➤ To configure My webMethods Server for Kerberos authentication

1. Log in to My webMethods Server as Administrator and configure LDAP for the Active Directory configured for Kerberos authentication server or create client user accounts in My webMethods Server. Even the user account for the Windows server machine must be included in LDAP or My webMethods Server user accounts.

For information about configuring LDAP, see [“Configuring an External LDAP, ADSI, or ADAM Directory Service” on page 112](#).

2. Edit and save the *Software AG_directory* \profiles\MWS_default\configuration\jaas.conf file to include the code below to the end of the file:

```
spnego-server{  
com.sun.security.auth.module.Krb5LoginModule required  
doNotPrompt=true  
principal="HTTP/<FQDN_of_Active_Directory_Server>"  
useKeyTab=true  
keyTab="<Keytab_file_absolute_path>"  
storeKey=true  
isInitiator=false  
debug=false;  
};
```

3. Edit and save the *Software AG_directory* \profiles\MWS_default\configuration\custom_wrapper.conf file to include the properties mentioned below:

```
wrapper.java.additional.602=-Dsun.security.krb5.debug=false  
wrapper.java.additional.603=-Djavax.security.auth.  
useSubjectCredsOnly=false
```

4. Restart My webMethods Server.
5. Log in to My webMethods Server as system administrator.
6. Navigate to **Configuration > KerberosAuthentication Administration** and provide the appropriate values for the Realm (specify all the machines managed by KDC) and the KDC server.

7. Navigate to **Configuration > Alias Management** and change the default authentication scheme for My webMethods Server to Kerberos. For instructions, see [“Specifying a Default Authentication Scheme” on page 264](#).
8. Restart My webMethods Server.

Configuring Web Clients to Use Kerberos Authentication

Configure Internet Explorer to Support Kerberos Authentication

➤ To configure Internet Explorer

1. In Internet Explorer, navigate to **Tools > Internet Options**.
2. Click **Security** and do the following:
 - a. Select **Local Intranet** zone.
 - b. Click **Sites**.
 - c. Clear the **Automatically detect intranet network** checkbox.
 - d. Select all these options:
 - **Include all local (intranet) sites not listed in other zones**
 - **Include all sites that bypass the proxy server**
 - **Include all network paths (UNCs)**
 - e. Click **Advanced** and ensure that the Windows Server address is listed in the **Websites** list.
For the Windows Server address, see [“Configuring Windows Server and Active Directory for Kerberos Authentication” on page 267](#).
 - f. Click **Close** and **OK** to return to the Security screen.
 - g. To set the security level, click **Custom level** and scroll to **User Authentication** in Security Settings, and select **Automatic logon only in Intranet zone**, if not already selected.
3. If you have a proxy server configured, include the server domain by doing the following:
 - a. In **Connections > LAN Settings > Proxy Server**, select **Use a proxy server for your LAN**.
 - b. Click **Advanced > Exceptions**, add Active Directory domain name.

Configure Mozilla Firefox to Support Kerberos Authentication

> To configure Mozilla Firefox

1. Enter `about:config` in the address bar and press Enter.
2. Scroll down to `network.negotiate-auth.trusted-uris` and double-click it.
3. Add or change the existing value to add the configured the Windows server domain name to the trusted URIs list. Use comma-separated list of URIs.

Configure Google Chrome to Support Kerberos Authentication

> To configure Google Chrome

1. Start Google chrome in command line by using `Chrome.exe --auth-server-whitelist="*.Active_Directory_Domain_Name"` command.

Configure Kerberos Authentication for Directory Services

You can configure a directory service in My webMethods Server to connect to the Active Directory using Kerberos. To configure this authentication option you must add your Kerberos realms in the `krb5.conf` file for the installation and add a new authentication module in the `jaas.conf` file for your server instance. To use Kerberos credentials cache, you must also configure the ticket cache location in the `jaas.conf` file.

To configure a directory service in My webMethods Server to connect to LDAP via Kerberos

1. Create a `krb5.conf` file in the `/conf/security` or `/jre/lib/security` directory of your Java installation. If a `krb5.conf` file is already available, edit it to include the realm configuration and the location of the Kerberos Key Distribution Center, as follows:

```
[libdefaults]
default_realm = MYREALM.COM
[realms]
MYREALM.COM =
{ kdc=myrealm.kdc_url:port }
```

2. Specify the location of the `krb5.conf` file by adding the following property in the `custom_wrapper.conf` file for My webMethods Server:

```
wrapper.java.additional.nnn=-Djava.security.krb5.conf="file_location/krb5.conf"
```

where `file_location` is the full path to your `krb5.conf` file.

3. Restart My webMethods Server.

4. Open the `jaas.config` file for your My webMethods Server instance in a text editor.

The file is located in `Software AG_directory/profiles/instance_name/configuration` directory.

5. At the bottom of the file, add one of the following snippets, depending on the required configuration:

- to register a new authentication module with name `MWSKerberosLDAP` that uses the credentials of the security principal for authentication:

```
MWSKerberosLDAP { com.sun.security.auth.module.Krb5LoginModule required; }
;
```

- to register a new authentication module with name `MWSKerberosLDAP` that uses Kerberos ticket cache:

```
MWSKerberosLDAP{
  com.sun.security.auth.module.Krb5LoginModule required useTicketCache=true
  ticketCache="path_to_ticket_cache_location/security_principal_krb5.tc"
};
```

For `ticketCache`, supply the full path and the file name of the credentials cache file for your security principal.

Note:

Do not change the module name!

6. In My webMethods, navigate to **Applications > Administration > My webMethods > Directory Services > Directory Services Administration**.
7. Click the name of the LDAP directory service you want to configure.
8. From the **Use Kerberos** drop-down list under **Connection Information**, select **Yes. Use Kerberos**.
9. Do one of the following:
 - For **Security Principal**, enter the name of the Kerberos principal, and for **Security Credentials**, enter the password of the Kerberos principal. These fields are required when you don't want to use Kerberos ticket cache.
 - Enable the **Use Ticket Cache** checkbox. For this setting to take effect, you must first configure a ticket cache location as described in step 2.
10. Click **Apply**.

Configuring NTLM Authentication

Using NTLM authentication, a user who has logged into a Windows domain does not have to re-authenticate to log in to a server. A Primary Domain Controller is a Microsoft Windows server

responsible for handling all accounts in a domain. To use NTLM authentication, you must explicitly specify the Primary Domain Controller in the NTLM Authentication Scheme.

➤ To specify a Primary Domain Controller for NTLM authentication

1. If you have NTLMv2 authentication configured, you need to disable it. For more information, see [“Disabling NTLMv2 Authentication” on page 273](#).
2. As system administrator, click **Administration Dashboard > Configuration > NTLM Authentication Administration**.
3. In the **Domain Controller Name** field of the Properties page, type the hostname of the Primary Domain Controller and click **Submit**.

Disabling NTLM Authentication

If you have NTLM enabled, you need to disable it before configuring NTLMv2.

➤ To disable NTLM Authentication

1. As system administrator, click **Administration Dashboard > Configuration > NTLM Authentication Administration**.
2. In the **Domain Controller Name** field, delete all characters including spaces and click **Submit**.

Configuring NTLMv2 Authentication

To use NTLM version 2 (NTLMv2), you need to purchase and install latest version of the Jespa Java software library from IOPLEX Software.

➤ To configure NTLMv2 authentication

1. If you have NTLM authentication configured, you need to disable it. For more information, see [“Disabling NTLM Authentication” on page 272](#).
2. Install and configure the Jespa Java software library, as described in the documentation provided when you download Jespa.

The Jespa library should be located here:

Software AG_directory /MWS/lib/

3. Stop My webMethods Server.
4. Extract jcifs-1.3.17.jar from wm_ntlmv2authadmin.pdp, which you can find at this location:

Software AG_directory /MWS/components/admin/configuration/ wm_ntlmv2authadmin.pdp

5. Copy `jespa-1.1.11.jar` from the Jespa Java software library and `jcifs-1.3.17.jar` to this location:
Software AG_directory /MWS/lib
6. From the command line, run `mws update`.
7. Restart My webMethods Server.
8. As system administrator, click **Administration Dashboard > Configuration > NTLMv2 Authentication Administration**.
9. In the **NTLMv2 Enabled** field, make sure **Yes, NTLMv2 is enabled** is set.
10. Configure the properties in the remaining fields according to the information provided in the Jespa documentation.
11. If you want to test the configuration, select **Yes, test configuration before save**.
12. Click **Submit**.
13. To make sure the new library is configured:
 - a. At a command line prompt, type the following command to move to the server's bin directory:

```
cd Software AG_directory\MWS\bin
```
 - b. Type `mws stop`.
 - c. Type `mws update`.
 - d. Type `mws start`.

Disabling NTLMv2 Authentication

If you have NTLMv2 enabled, you need to disable it before configuring NTLM.

> To disable NTLMv2 Authentication

1. As system administrator, click **Administration Dashboard > Configuration > NTLMv2 Authentication Administration**.
2. In the **NTLMv2 Enabled** field, make sure **No, NTLMv2 is disabled** is set and click **Submit**.

3. Delete the Jespa and Java CIFS libraries from this location:

Software AG_directory /MWS/lib/ext

4. To de-reference the libraries and reconfigure the server:

- a. At a command line prompt, type the following command to move to the server's bin directory:

```
cd /Software AG_directory\MWS\bin
```

- b. Type `mws stop`.
- c. Type `mws update`.
- d. Type `mws start`.

Configuring External Configuration Credentials

The HTTP Header Authentication Administration page allows system administrators to configure My webMethods Server to accept external HTTP authentication credentials from third party security and access control products such as SiteMinder (Computer Associates) or Oblix. These credentials are case-sensitive and, depending on the platform and web server, will most likely be `sm_user` or `SM_USER`.

Enabling Authentication

Important:

The HTTP Header Authentication Administration page should only be enabled if you are using a third-party security provider. After the page is enabled, the server acts as though all users have been authenticated.

➤ **To accept authentication from a third party security and access control product**

1. As system administrator, click **Administration Dashboard > Configuration > HTTP Header Authentication Administration**.
2. For **User Header Name**, type `sm_user` or `SM_user`.
3. Select the **Enable HTTP Header Authentication** check box.
4. If appropriate, in the **Logout URL** field, type the URL to which the user is redirected after logging out of the server.
5. Click **Submit**.

6. Configure the third party security and access control software as directed in your vendor's product documentation.

Note:

To properly configure an external security and access control product, both My webMethods Server and the third party product *must* point to the same directory server instance.

Checking Logs for HTTP Header Authentication Problems

If you are having a problem in getting HTTP Header authentication on to work properly, you can check log files to assist in diagnosing the problem. Log messages for HTTP Header authentication are assigned to the portalLogin category. Before you can display HTTP Header authentication logging messages, you need to change the logging threshold values. The default thresholds for writing to the console, the _full.log file, and the portalLogin.log file are set to the INFO log level but HTTP Header authentication logging messages use the DEBUG log level, which is lower.

Server log files reside in the *Software AG_directory \MWS\server\serverName\logs* directory. For information on controlling the collection of logs, see [“Viewing Logging Messages” on page 318](#). For information on searching for log messages, see [“Viewing Logging Messages” on page 318](#).

Setting Login Logging Thresholds

You need to set both the category and output settings to DEBUG if you want the logging messages to be written to the output. For information on setting logging thresholds, see [“Controlling Server Logging” on page 306](#).

➤ To set category and output thresholds for HTTP Header authentication

1. As system administrator, click **Administration Dashboard > Analysis > Logging Configuration > Logging Thresholds**.
2. In the **Category Threshold** list, select the DEBUG log level for any or all of the following logging categories:

The following table lists the available logging categories:

Logging category	Description
root	Controls the output for the console and the _full.log file
portalLogin	Controls the output for the portalLogin.log file

3. In the **Output Threshold** list, select the DEBUG log level for any or all of the following logging output types:

The following table lists the available logging outputs:

Logging output	Description
Console	Controls the output for logging messages sent to the console
_full.log	Controls the output for logging messages sent to the _full.log file
portalLogin	Controls the output for logging messages sent to the portalLogin.log file

4. Click **Apply**.

Checking HTTP Header Authentication Logs for Problems

With HTTP Header authentication enabled, the server acts as though all users have been authenticated. With this in mind, the log messages will reveal one of three likely outcomes, as described in the following sections.

The Login is Successful

Messages for a successful login using HTTP Header authentication look similar to the following example:

```
Date_and Time (portalLogin : DEBUG) - HttpHeadersHandler Auth Handler
```

```
looking for: user_name
```

```
Date_and Time (portalLogin : DEBUG) - Found userID: user_name
```

where *user_name* is the name of the user who logged in under HTTP Header authentication.

HTTP Header Authentication is Disabled

If you have not enabled HTTP Header authentication, the log message looks similar to the following example:

```
Date_and Time (portalLogin : DEBUG) - HttpHeadersHandler Auth Handler is
```

```
not enabled
```

To enable HTTP Header authentication, see [“Enabling Authentication” on page 274](#).

The Problem Rests with the Third-Party Site

If the third-party site is not configured correctly, HTTP Header authentication will fail. The resulting log message looks similar to the following example:

```
Date_and Time (portalLogin : DEBUG) - HttpHeadersHandler Auth Handler
```

```
looking for:
```

```
Date_and Time (portalLogin : DEBUG) - No value found!
```

Configuring My webMethods Server Single Sign-On

Single sign-on (SSO) enables a user to log into one application and then use other applications without having to log into each one separately. My webMethods Server supports single sign-on through the Security Assertion Markup Language (SAML), an XML-based framework for the exchange of security information. Using SAML, an entity on a target computer grants access based on an assertion from the source computer that the user is logged into the source computer.

With SAML 1.0, My webMethods Server can provide a single sign-on capability in the following ways:

- Between a source server and one or more target servers
- Between a server and other webMethods applications that have single sign-on capability
- Between a server and a third-party application that supports SAML
- (Deprecated) Between a server, an Artifact Receiver that authenticates the user sign-on, and a target web application

Using this model, one server is the source, providing a central login for users. Links on pages on the source server point to any number of SAML-capable entities. Also, a target server can accept assertions from any number of servers as long as the truststore of the target server has the certificate of the source server.

To take advantage of single sign-on, a user must be known on both the source server and the target entity. In most cases, common knowledge of a user is provided by use of the same directory service.

With SAML 2.0, you can configure My webMethods Server to authenticate users, registered with a third-party identity provider, using both Identity Provider (IDP) initiated and Service Provider (SP) initiated SSO flow.

To configure any of the supported SSO flows, you must add the certificate used in signing the assertion to the truststore of the target My webMethods Server instance. For more information, see [“Importing CA Certificates” on page 32](#).

Configuring a Server as a Target for Single Sign-On

A server can be a target for only one single sign-on source at a time.

➤ To configure a server to be a target for single sign-on

1. As system administrator, click **Administration Dashboard > Configuration > SAML Authentication Administration**.
2. On the **Properties** page, specify:

The following table lists the properties, required to configure a server as a target for single sign-on:

Property	Description
Artifact Parameter Name	(Deprecated) If this is a SAML connection with another webMethods server, do not change the default value SAMLart. If this is a SAML connection to a third-party source, type the artifact parameter name used by the third-party application.
Assertion Parameter Name	The HTTP request parameter name where the server will look for the SAML assertion value. The default value is SAMLResponse.
Security Provider URI	(Deprecated—used with the Artifact Parameter Name) Type the URI of the SAML security provider (source). If this is a connection with another webMethods server, use this syntax: <code>server:port/services/SAML</code> where <i>server</i> is the host where the source server is running and <i>port</i> is the server port number. The default port number is 8585.

3. Click **Submit**.

Setting SAML Links on a Source Server

On any page, you can add a link to a SAML target entity, such as a server. If the target accepts SAML assertions from the source server, when a known user clicks the link, no login credentials are required. If the target entity does not accept SAML assertions from the source server, or if the user is not known on the target entity, login credentials may be required.

(Deprecated—valid only with a SAML Artifact Receiver) Under the SAML specification, an intermediary called an artifact receiver can perform authentication on behalf of the target web application. In such a case, the SAML source requires two URLs: one for the Artifact Receiver and one for the target web application. You can place one or more SAML links on any page you have permission to edit.

You can place one or more SAML links on any page you have permission to edit.

➤ To create a SAML link on a source page

1. In the upper right-hand corner of the page, click  and click **Edit Page**.
2. In the **Root** list of the **Available Portlets** panel, click **Links**.
3. In the **Links** list of the **Available Portlets** panel, drag the **wm_xt_ssolink** portlet and drop it onto the page at the location where you want to add the link.

A red box appears beneath the cursor location whenever the cursor is over a valid page location, indicating where the portlet would be positioned if you released the mouse button.

4. On the left side of the page control area, click **Save**.

5. At the right edge of the title bar for the single sign-on portlet, click  and click **Properties**.
6. On the **Properties** page, specify:

The following table lists the properties to modify when configuring SAML links:

Field	Description				
Name	Replace <code>wm_xt_ssolink</code> with the text that is to go with the link.				
SAML Type	Select the version of the SAML specification to be used: <ul style="list-style-type: none"> ■ SAML2 POST ■ SAML1 POST ■ SAML Artifact — (Deprecated) 				
SAML Authentication URL	Type the URL for a resource on the target computer. The target can be any page on a server. (Deprecated) If you are connecting to a web application through a SAML Artifact Receiver, use this field for the Artifact Receiver URL.				
Use POST or GET	(Deprecated — Valid only if the SAML Type field is set to SAML Artifact) Determines the method used to pass data to the target computer. <table border="0"> <tr> <td style="padding-right: 20px;">POST</td> <td>Passes data to a gateway program's STDIN. POST, the default, is the preferred method for single sign-on data.</td> </tr> <tr> <td>GET</td> <td>Passes data as a string appended to the URL after a question mark.</td> </tr> </table>	POST	Passes data to a gateway program's STDIN. POST, the default, is the preferred method for single sign-on data.	GET	Passes data as a string appended to the URL after a question mark.
POST	Passes data to a gateway program's STDIN. POST, the default, is the preferred method for single sign-on data.				
GET	Passes data as a string appended to the URL after a question mark.				
Assertion Parameter Name	The HTTP request parameter name where the server will look for the SAML assertion value. The default value is <code>SAMLResponse</code> .				
Artifact Parameter Name	(Deprecated — Valid only if the SAML Type field is set to SAML Artifact) If this is a SAML connection with another server or other webMethods product, do not change the default value <code>SAMLart</code> . If this is a SAML connection to a third-party source, type the artifact parameter name used by the third-party application.				
Application Target URL	(Deprecated) If you have typed the URL for a SAML Artifact Receiver in the SAML Authentication URL field, type the URL for a web application. Otherwise, leave this field empty.				

7. Click **Apply**.

Using Single Sign-On with SAML and a Third-Party Identity Provider

The following high-level steps apply when My webMethods Server authenticates users that are not present in any of the available directory services, and are registered only with a trusted identity provider:

For Identity Provider Initiated SSO:

- A user that is already authenticated with the IDP attempts to access a protected My webMethods Server resource.
- The IDP redirects the user with the authentication response to My webMethods Server and sends a SAML response token as a POST parameter to My webMethods Server using SAML POST binding.
- My webMethods Server validates the SAML response based on the signature details in the SAML response. The signature on the assertion is validated using the public key of the identity provider available in the metadata file.
- My webMethods Server processes the SAML response and verifies the user details present in the token before serving the requested content.

For more information about configuring IDP initiated SSO, see [“Configuring Identity Provider Initiated Single Sign-On with a Third-Party Identity Provider” on page 281](#).

For Service Provider Initiated SSO:

- A user that is registered with the IDP provider requests access to a protected My webMethods Server resource.
- My webMethods Server sends a SAML request for authentication through the browser to the SSO service of the IDP.
- If the user is not logged on to the IDP, the IDP asks for credentials (for example ID and password) and the user logs on.
- The SSO service returns an HTML form to the browser and includes the SAML response with the authentication assertion. The browser posts the HTML form back to My webMethods Server to verify the user details and serve the content.

For more information about configuring SP initiated SSO, see [“Configuring Service Provider Initiated Single Sign-On with a Third-Party Identity Provider” on page 281](#).

Configuring Identity Provider Initiated Single Sign-On with a Third-Party Identity Provider

➤ To configure IDP Initiated SSO using a third-party IDP

1. Ensure that My webMethods Server is configured to use an HTTPS port.
2. Set the required properties in the `websso.properties` file. For information about working with the `websso.properties` file, see [“Setting Properties in the websso.properties File” on page 282](#).
3. Import the IDP certificate to the My webMethods Server truststore using the `keytool` command of the JVM. For more information, see [“Importing CA Certificates” on page 32](#).

4. Start My webMethods Server.

On startup, My webMethods Server creates two metadata files in the *Software AG_directory* \MWS\server\serverName\config directory: `SPMetadata.xml` and `IDPMetadata.xml`.

5. Register My webMethods Server as a service provider with the external identity provider using the information in the *Software AG_directory* \MWS\server\serverName\config\SPMetadata.xml file, or copy the file to the required location on the IDP sever.

The identity provider uses the endpoint location of the My webMethods Server instance from the `SPMetadata.xml` file to list My webMethods Server as a service provider.

6. When the IDP provider is configured to send encrypted assertions, replace the following (default) JCE policy files in *Software AG_directory* \jvm\operating_system\jre\lib\security folder with the latest JCE files:

- `local_policy.jar`
- `US_export_policy.jar`

7. Restart My webMethods Server.

Configuring Service Provider Initiated Single Sign-On with a Third-Party Identity Provider

➤ To configure SP Initiated SSO using a third-party IDP

1. Ensure that My webMethods Server is configured to use an HTTPS port.
2. Set the required properties in the `websso.properties` file. For information about working with the `websso.properties` file, see [“Setting Properties in the websso.properties File” on page 282](#).

3. Import the IDP certificate to the My webMethods Server truststore using the `keytool` command of the JVM. For more information, see [“Importing CA Certificates” on page 32](#).
4. Start My webMethods Server.
5. Register My webMethods Server as a service provider with the external identity provider using the information in the *Software AG_directory* \MWS\server\serverName\config\SPMetadata.xml file, or copy the file to the required location on the IDP sever.

The identity provider uses the endpoint location of the My webMethods Server instance from the `SPMetadata.xml` file to list My webMethods Server as a service provider.

6. Import the My webMethods Server certificate to the IDP truststore. For more information, see the identity provider documentation.
7. As sysadmin, go to the **SAML Authentication Administration** portlet and enable SP initiated SSO.
8. Restart My webMethods Server.

Setting Properties in the `websso.properties` File

You set the properties required for single sign-on using a third-party identity provider (IDP) in the `websso.properties` file.

➤ To set properties in the `websso.properties` file

1. At a command line prompt, type the following command to access the bin directory of the server:

```
cd Software AG_directory\MWS\bin
```

2. Type the following command to retrieve the `websso.properties` file from the My webMethods Server database:

```
mws.{sh|bat} -s serverName getconfig websso.properties
```

3. Open the `websso.properties` file in a text editor. You can find the file at the following location:

```
Software AG_directory \MWS\server\serverName\config\
```

4. Set the properties in the `websso.properties` file. For more information about each property, see [“Property Setting for Single Sign-On Using a Third-Party IDP” on page 283](#).
5. Type the following command to save the `websso.properties` file in the My webMethods Server database:

```
mws.{sh|bat} -s serverName putconfig websso.properties
```

6. Delete the websso.properties file from the following location:

Software AG_directory \MWS\server*serverName*\config\

If you do not delete the file, the server will continue to use the local version of the websso.properties file at this location.

7. Restart My webMethods Server.

Property Setting for Single Sign-On Using a Third-Party IDP

In the *Software AG_directory* \MWS\server*serverName*\config\websso.properties file, provide the values required for single sign-on using a third-party IDP.

The following table lists the properties, required to configure single sign-on with a third-party identity provider in My webMethods Server:

Property	Description
SSO_KEYSTORE	The keystore used for SSO communication using SAML2.0. My webMethods Server stores the keystores in <i>Software AG_directory</i> \MWS\server\ <i>serverName</i> \config\security directory. The value of SSO_KEYSTORE can be an absolute path or a path relative to the config directory. The default keystore, localhost.p12, is present in the config\security directory.
SSO_KEYSTORE_PASSWORD	The keystore password. The keystore password can be in plain text or encrypted. For information about password encryption, see “Generating an Encrypted Password” on page 35 .
SSO_KEYSTORE_TYPE	The keystore type. It can be JKS or PKCS12.
SSO_SIGN_ALIAS	The alias name to be used for signing the SAML response.
SSO_SIGN_ALIAS_PASSWORD	The password for signing alias.
SSO_ENCRYPT_ALIAS	The alias name to be used for SAML response encryption.
SSO_ENCRYPT_ALIAS_PASSWORD	The password for alias encryption.
SSO_DEFAULT_ALIAS	The default alias name in case the signing alias (SSO_SIGN_ALIAS) and encryption alias

Property	Description
	<p>(SSO_ENCRYPT_ALIAS) are same. If you specify a value for SSO_DEFAULT_ALIAS, then the password for default alias is assumed to be same as the keystore password.</p> <p>If you specify signing alias and encryption alias, you need not specify the default alias. If you specify a value for SSO_DEFAULT_ALIAS, the values set for the following properties are ignored:</p> <ul style="list-style-type: none"> ■ SSO_SIGN_ALIAS ■ SSO_SIGN_ALIAS_PASSWORD ■ SSO_ENCRYPT_ALIAS ■ SSO_ENCRYPT_ALIAS_PASSWORD
SSO_IDP_METADATA_URL	The URL of the Identity Provider's metadata file.

Example Property Settings in websso.properties File

```
SSO_KEYSTORE=config/security/localhost.p12
SSO_KEYSTORE_PASSWORD={AES}Y5IgMqjfvkbg7p5VUZztw\=\=
SSO_KEYSTORE_TYPE=PKCS12
SSO_SIGN_ALIAS=localhost
SSO_SIGN_ALIAS_PASSWORD={AES}Y5IgMqjfvkbg7p5VUZztw\=\=
SSO_ENCRYPT_ALIAS=localhost
SSO_ENCRYPT_ALIAS_PASSWORD={AES}Y5IgMqjfvkbg7p5VUZztw\=\=
SSO_DEFAULT_ALIAS=localhost
SSO_IDP_METADATA_URL=
"http://example.org/metadata.xml"
```

Additional Configurations for SAML SSO

When logged in as sysadmin, use the **SAML Authentication Administration** portlet to configure the following additional settings for SAML SSO:

Field	Description
SP Initiated SSO Enabled	Required. Enables or disables SP Initiated SSO. The default value is No. Disable SP Initiated SSO.
Include login form	Required. Whether to include a simple login form for basic authentication together with the SSO link, and allow the user to supply different credentials. The default value is Yes. Include login form.

Field	Description
Create User	Required. Whether to register a new user in the system directory if the user does not exist. The default value is <code>false</code> .
Return URL at logout	The page to display when a user, authenticated using SSO logs out from My webMethods. Must point to a whitelisted domain. For more information about adding servers to the My webMethods Server whitelist, see “Adding Servers to a Whitelist” on page 28.
Role Name	The name of the role to be injected with custom SAML attributes. The default value is <code>SamLSinkRole</code> .
Role Member Attributes	The list of SAML attributes to add to the membership attributes of the role. The list must be comma-separated. The default value is <code>nameId</code> .

Configuring OAuth 2.0 Authentication

With the OAuth 2.0 authentication scheme, users can log in to My webMethods using accounts from Google, Twitter, Salesforce, or another identity provider that supports the OAuth 2.0 and Open ID Connect protocols and exposes a discovery service. You can control what server resources, or layered products the users can access, based on user roles and role attributes. You configure the user roles to accept claims, submitted by the identity provider to determine user membership dynamically at login. You can configure multiple OAuth 2.0 services for different identity providers in My webMethods Server.

Users and Roles for External Accounts

By default, My webMethods Server creates a new system user for each external account, that logs in using the OAuth 2.0 authentication scheme. These system users are assigned to a default role named `OAuthSinkRole`. You can create and configure a custom role to use for assigning OAuth 2.0 authenticated users. For more information about My webMethods Server roles, see [“Managing Roles and Access to My webMethods”](#) on page 183.

Optionally, you can also create a custom service that registers internal users for the external user accounts. For more information, see *webMethods CAF and My webMethods Server Java API Reference*.

Configuring an Authentication Flow over OAuth 2.0 and OpenID Connect

To enable users to log in to My webMethods with credentials from third-party identity providers:

1. Register My webMethods Server with the authorization server/identity provider. For more information, see [“Registering My webMethods Server with an Identity Provider”](#) on page 286.
2. Configure an OAuth 2.0 service in My webMethods Server. For more information, see [“Configuring an OAuth 2.0 Service”](#) on page 286.
3. Customize the My webMethods login page, or create a custom login page that redirects users to the identity provider. For more information, see [“Customizing the My webMethods Login Page for OAuth 2.0 Authentication”](#) on page 288.

Registering My webMethods Server with an Identity Provider

See the official documentation of your authorization server/identity provider for information how to register My webMethods Server as a client application. You will need the following information from your registration with the identity provider to configure an OAuth 2.0 service in My webMethods Server later:

- The URL of the discovery service of your OpenID Connect Provider
- The OAuth 2.0 client identifier for My webMethods Server
- The OAuth 2.0 client secret for My webMethods Server
- The redirection URLs, preregistered at the OpenID Connect Provider
- The access claim that allows registering users in My webMethods Server

Configuring an OAuth 2.0 Service

Both Administrator and sysadmin users can add, remove, or modify OAuth 2.0 services.

> To configure a new OAuth 2.0 service

1. Navigate to the **OAuth2 Administration** page and click the **Add OAuth Configuration** tab.
 - As sysadmin: **Folders > My webMethods Applications > Fabric Tasks > Administration > My webMethods > OAuth2 Administration**
 - As Administrator: **Applications > Administration > My webMehods > OAuth2 Administration**
2. On the **Add OAuth Configuration** tab, specify:

The following table lists the properties, required to configure an OAuth service:

Field	Description
Name	Required. The name of the OAuth 2.0 service.

Field	Description
Service Enabled	Required. Select an option from the drop-down list to enable or disable the OAuth 2.0 service. By default, newly created services are enabled.
Discovery Document URL	Required. The URL of the discovery service of the OpenID Connect provider, from your registration with the provider.
OAuth 2.0 Client Identifier	Required. The OAuth 2.0 client identifier, valid at the authorization server, from your registration with the provider.
OAuth 2.0 Client Secret	Required. The client secret to use for OAuth 2.0 authorization, from your registration with the provider.
OpenID Connect Scopes	Required. The scope of the requested authorization, as defined by OpenID Connect. The default is <code>openid,profile,email</code> . For more information about available scopes, see the identity provider documentation.
Redirection URI	Required. The My webMethods Server URL that you provided when registering with the identity provider.
OpenID Connect Provider Name	Required. The name of the OpenID Connect provider. My webMethods Server displays this name on the preconfigured Login with <i>provider_name</i> button. For more information, see “Customizing the My webMethods Login Page for OAuth 2.0 Authentication” on page 288 .
Access Claim	Required. The access claim that allows registering users in My webMethods Server. For more information about available claims, see the identity provider documentation.
Subject Claim	Required. The subject claim that identifies the user.
User Service Name	Optional. The name of a custom service that creates internal My webMethods users for the external accounts, authenticated using the OAuth 2.0 flow. For more information about users and roles in the OAuth 2.0 authentication flow, see “Configuring OAuth 2.0 Authentication” on page 285 .
Role Name	Optional. The name of the role to inject with custom OAuth attributes. My webMethods users that authenticate using the OAuth 2.0 flow will be assigned to this role. The default is OAuthSinkRole .

Field	Description
Role Member Attributes	Optional. The list of claims to add to the membership attributes of the role. Specify a comma-separated list of claims. For more information about available claims, see the identity provider documentation.
Prompt	Optional. The type of prompt that the identity provider uses to authenticate a user. The default value is <code>login</code> - the identity provider asks the user to log in.
Create New User	Optional. Whether to register a new My webMethods system user for each user that logs in using the OAuth authentication flow. The default value is Yes. Create a new user.

3. Click **Submit**.

When you submit the configuration, My webMethods Server generates an authentication URL in the **Auth URL** field. This URL is required to add the OAuth 2.0 configuration as a login option. For more information about adding OAuth 2.0 authentication to the My webMethods login page, see [“Customizing the My webMethods Login Page for OAuth 2.0 Authentication” on page 288](#).

Customizing the My webMethods Login Page for OAuth 2.0 Authentication

You can add a login button with the name of the external provider to the default login options on the My webMethods. The provider name on the button is the same as the **OpenID Connect Provider Name** that you specified on the **Add OAuth2 Configuration** tab.

> To add an OAuth 2.0 login option

1. As sysadmin, navigate to **Folders > System > Portlets > Administration > OAuth Provider Configurations**.
2. Click **Tools > Edit page**, and then drag the Login portlet to the **OAuth Provider Configurations** page.
3. If required, modify the portlet preferences and save your changes.

Modify the login page rules to display the OAuth Configuration to which you added a login portlet as the default My webMethods login page. For more information about login page rules, see [“Creating Login Page Rules” on page 351](#).

Clearing Session Passwords from Memory

By default, when a user logs in, the password is stored until the user logs out or until the session times out. You can, however, cause My webMethods Server to clear passwords from memory immediately after the login is completed. This setting clears all passwords presented to My webMethods Server; you cannot clear passwords on a case-by-case basis.

➤ **To clear session passwords from memory**

1. As system administrator, click **Administration > Folders > System > Managers > sessionManager > default > validate**.
2. At **Configuration XML**, click **Edit**.
3. In the Edit Text Area, change this text:

```
clearPassword= false
```

to this:

```
clearPassword= true
```

4. To save the file and close the editor, click **Update**.
5. Click **Apply**.

This setting remains until you change it, even if you stop and restart My webMethods Server.

Retaining Session Passwords in Memory

➤ **To retain session passwords in memory**

1. As system administrator, click **Administration > Folders > System > Managers > sessionManager > default > validate**.
2. At **Configuration XML**, click **Edit**.
3. In the Edit Text Area, change this text:

```
clearPassword= true
```

to this:

```
clearPassword= false
```

4. To save the file and close the editor, click **Update**.

5. Click **Apply**.

Turning On or Off Auto Complete for Usernames and Passwords

When a user logs on to My webMethods Server, the browser allows the user to remember the username and password. You can configure the My webMethods Server login portlet to turn on or off the auto complete functionality. By default, auto complete is turned on.

> To turn on or off auto complete for usernames and passwords

1. As system administrator, go to **Folders > My webMethods Applications > webMethods Application Data > My webMethods Login Page**.
2. Append `?layout=details` to the end of the My webMethods Server URL in the browser address bar to display the login portlet.
3. Click  for the login portlet and select **Properties**.
4. Do one of the following:
 - To turn off auto complete for usernames and passwords, clear the **Allow Auto Complete** check box.
 - To turn on auto complete for usernames and passwords, select the **Allow Auto Complete** check box. The check box is selected by default.
5. Click **Apply**.

Controlling the Number of Failed Login Attempts

My webMethods Server controls the number of failed login attempts it will allow before it temporarily locks the user account. On the **Failed Logins Administration**, system administrators can configure the maximum number of failed login attempts for My webMethods users, and the period of time for which My webMethods Server suspends subsequent login attempts.

Changes on the **Failed Logins Administration** page affect all logins to the server.

> To control the number of failed login attempts

1. As system administrator, click **Administration Dashboard > Configuration > Failed Logins Administration**.
2. In the **Configure Failed Logins Behavior** section, specify:

The following table lists the properties to configure to control the number of failed login attempts:

Property	Description
Window Duration	The time interval in milliseconds between subsequent checks for failed login attempts. The default is <i>30000</i> .
Max Attempts in Window	The maximum number of login attempts during the window duration. The default is <i>10</i> .
Lockout Duration	The time interval in milliseconds for which the user will be locked out of the server. The default is <i>30000</i> .

3. Click **Apply**.

Controlling Login IP Ranges

You can use the Lockdown portlet to control the IP ranges from which users are allowed to log in to My webMethods Server. Up to four different IP ranges are allowed. The portlet does not take into account the credentials of the person logging in or any authentication mechanism. Upon being unable to log in, the user is redirected to an error page of your choosing. This portlet is useful for a site that allows guest access only, or one that is to be access only from a secure location.

Important:

If you make an error in configuring IP ranges, you may not be able to log in to correct the problem. To correct the problem, you need to have physical access to the machine on which My webMethods Server is running.

Deploying the Lockdown Portlet

The Lockdown portlet is not deployed automatically when you initialize My webMethods Server. The portlet is included in a My webMethods Server installation, but you must deploy it before use.

➤ To deploy the Lockdown portlet

1. Locate the Lockdown portlet within the webMethods installation directory at this location:

```
\ Software AG_directory \MWS\components\extras\security\wm_lockdown.pdp
```

2. Copy the `wm_lockdown.pdp` file to the deploy directory for the server instance:

```
\ Software AG_directory \MWS\server\serverName\deploy
```

My webMethods Server detects the presence of the portlet and automatically deploys it.

Configuring the Lockdown Portlet

After deploying the Lockdown Portlet to My webMethods Server, you can modify the portlet configuration.

> To configure the Lockdown portlet

1. As system administrator, click **Administration > Folders > System > Portlets > Administration > Portal Lockdown Administration**.
2. For **Error page** property, select the page to redirect users to when their login attempt fails, by clicking one of the following buttons:
 - **Browse** - in the **Select Portal Resource** dialog window, move the target page to the **Selected Items** box and click **Select**.
 - **Use Alias** - in the **Alias Name** field, type the alias of the target page. Click **Test** to determine if the alias is valid and the alias target is the correct one. If the alias is correct, click **Select**.
3. (Optional) In the **E-mail Address for Login Notification** field, type an email address to which My webMethods Server should send a notification if a login is attempted from a disallowed IP address.
4. Specify up to four IP ranges from which users are allowed to log into My webMethods Server.
For each range, provide a value in the **Start IP Range** and **End IP Range** fields.
5. On the **Is Active** field, choose the **True** option.
6. Click **Submit**.

The lockdown takes effect immediately.

Disabling the Lockdown Portlet in My webMethods Server

If you can log in to My webMethods Server as system administrator, you can disable the Lockdown portlet by doing the following.

> To disable the Lockdown portlet in My webMethods Server

1. As system administrator, click **Administration > Folders > System > Portlets > Administration > Portal Lockdown Administration**.
2. On the **Is Active** field, choose the **False** option.
3. Click **Submit**.

Disabling the Lockdown Portlet Manually

If you cannot log in to My webMethods Server as system administrator, you need to have physical access to the machine on which My webMethods Server is installed, and modify the `portal.properties` file. Everything in this file is set as a system property, read after the My webMethods Server starts but prior to initialization of portlets.

➤ To disable the Lockdown portlet manually

1. On the machine where My webMethods Server is installed, locate this file:

```
\ Software AG_directory \MWS\server\serverName\config\portal.properties
```

2. Open the `portal.properties` file in an editor and add the following line at the end of the file:

```
lockdown.disable=false
```

For example, with a descriptive comment included:

```
#=====
```

```
# Lockdown Portlet
```

```
lockdown.disable=false
```

3. Restart My webMethods Server.

Note:

After you have corrected the IP address range problem, you cannot reactivate the Lockdown portlet until you remove this line from the `portal.properties` file.

Encrypting Passwords for Global Environment Variables

You can encrypt password fields for custom entries in Global Defaults environment variables.

➤ To encrypt password fields for Global Defaults environment variables

1. As system administrator, click **Administration Dashboard > Configuration > CAF Application Runtime Configuration**.
2. In the result titlebar, click .
3. In the **Password Env-Entry Names** field, type the name of a custom environment entry after the existing entries, separated by a comma, and click **Save**:

For example, `wsclient-password,jcr/systemPassword,My_Password`

4. Click **Configure Global Defaults**.
5. In the tree view, click **Environment Entries**, and then click **Add New Entry**.
6. In the **Name** field, type the name of the custom environment entry (for example, `My_Password`) and click **Add It**.
7. Type a password value in the newly created environment entry and click **Apply**.

If you have correctly configured the environment entry, the password is masked as you type it into the field. In addition, the password value is encrypted before it is stored in the database.

If the consumer of the password value needs to get the original value back (for example, to log in to some external server) it is possible to use the `CipherUtil.decrypt(value)` Java API to get the original value back. For information about the `CipherUtil` class, see *webMethods CAF and My webMethods Server Java API Reference*.

Allowing Context Impersonation

There may be cases in which you want to allow a user to impersonate another user. For example, impersonation is used by the default Java Content Repository (JCR) Client. The JCR session is created as a system user (SysAdmin by default) with a known password and then the JCR session uses impersonation to switch to the real current user. This impersonation makes it possible for the current user to log into the JCR repository.

We do not recommend that you use the SysAdmin user to implement context impersonation, but another way to do so is to create a user with the Impersonate Users Functional Privilege set.

To set the Impersonate Users Functional Privilege for a user, see [“Managing Access Privileges and Functional Privileges” on page 173](#).

On the Permissions panel, Impersonate Users is located at **Functional Privileges > MWS > Impersonate Users**.

Using Password Complexity Policies

A password complexity policy enforces requirements that make user passwords more resistant to brute-force attacks. You can create a password complexity class and add it to My webMethods Server for use with the system directory service. You cannot use this function for external directory services.

My webMethods Server includes an out-of-the-box password complexity policy class, the `DefaultSystemPasswordComplexityPolicy`, which is not enabled by default. The default password complexity enforces the following requirements to all system user passwords:

- Minimum password length of eight symbols;
- Maximum password length of 64 symbols;

- Maximum three identical or sequential characters in a row, for example aaa or 123.

You can enable the default password complexity policy, or configure your custom password complexity implementations as follows:

➤ **To configure a password complexity policy for My webMethods Server**

1. Log in to My webMethods Server and go to the **Properties** page of the `system` directory service:
 - As SysAdmin: **Administration Dashboard > User Management > Directory Services Administration**
 - As My webMethods Administrator: **Navigate > Applications > My webMethods > Directory services**
2. On the **Directory Services** page, click the **system** directory service.
3. In the **Password Complexity Class** field under **Security Information**, select one of the following:
 - `com.webmethods.portal.service.dir.impl.DefaultSystemPasswordComplexityPolicy` - to use the built-in option for password complexity.
 - `other` - to supply the fully qualified name of the class that contains your custom password complexity policy.
4. Click **Apply**.

Adding Custom Password Complexity Policies

To implement a custom password complexity policy, you include a password complexity Java class in a Composite Application Framework (CAF) application. When you deploy the application to My webMethods Server, the server registers the Java class. The password complexity policy component is then available for use during subsequent requests to the server. Implementing the password complexity policy does not require restarting My webMethods Server.

Before creating the Java class, you must create a CAF application in Designer. For more information about creating, working with, and deploying CAF applications, see *webMethods CAF and OpenUI Development Help*.

➤ **To implement a custom password complexity as a Java class**

1. In an existing CAF application in Designer, create a Java class that implements the `ISystemPasswordComplexityPolicy` interface. The component is registered as an OSGi service via the `@Component` annotation.

The following example shows how to implement the custom password complexity policy.

```
package caf.war.cafapp1.dir;
```

```

import java.util.Map;
import java.util.regex.Pattern;

import org.osgi.service.component.annotations.Activate;
import org.osgi.service.component.annotations.Component;
import org.osgi.service.component.annotations.Deactivate;
import org.osgi.service.component.annotations.Reference;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

import com.webmethods.portal.resources.Ui;
import com.webmethods.portal.service.dir.IDirPrincipal;
import com.webmethods.portal.service.dir.ISystemPasswordComplexityPolicy;
import com.webmethods.portal.service.global.IGlobalProvider;
import com.webmethods.portal.service.global.IMessageInfo;
import com.webmethods.portal.system.init.InitializationException;

/**
 * Custom implementation of password complexity policy
 */
@Component (
    service = {
        ISystemPasswordComplexityPolicy.class
    },
    property = {
        "name:String=CustomSystemPasswordComplexityPolicy1"
    },
    immediate = true
)
public class CustomSystemPasswordComplexityPolicy1 implements
    ISystemPasswordComplexityPolicy {
    private Logger logger = LoggerFactory.getLogger(getClass());

    //references to other services
    private IGlobalProvider gp = null;

    /**
     * Reference injection of the other OSGi service
     */
    @Reference
    protected void bindGlobalProvider(IGlobalProvider globalProvider) {
        if (logger.isTraceEnabled()) {
            logger.trace("Bound IGlobalProvider");
        }

        this.gp = globalProvider;
    }

    /**
     * Activation of OSGi declaritive service
     * @param config the configuration properties of the OSGi service
     * @throws InitializationException if activation fails
     */
    @Activate
    protected void activate(Map<String, Object> config) throws
    InitializationException {
        if (logger.isInfoEnabled()) {
            logger.info("Activating service");
        }
    }
}

```

```

/**
 * Deactivation of OSGi declaritive service
 */
@Deactivate
protected void deactivate() {
    if (logger.isInfoEnabled()) {
        logger.info("Deactivating service");
    }
}

/**
 * Check the candidate password to make sure the value satisfies the
complexity
 * requirements. If the password is not complex enough, this method
should throw
 * an {@link InvalidPasswordException} with the reason. If no exception
is thrown
 * the password is valid.
 *
 * @param candidatePassword the password to check
 * @throws InvalidPasswordException if the password is not valid
 */
@Override
public void checkPasswordForNewUser(String candidatePassword)
    throws InvalidPasswordException {
    if (candidatePassword == null || candidatePassword.length() < 6) {
        IMessageInfo messageInfo = gp.getMessageInfo(Ui.class,
            "password.too.short", null);
        String msg = messageInfo.getLocalizedMessage(false);
        throw new InvalidPasswordException(msg);
    }

    boolean hasUpper =
!candidatePassword.toLowerCase().equals(candidatePassword);
    boolean hasLower =
!candidatePassword.toUpperCase().equals(candidatePassword);
    if (!(hasUpper && hasLower)) {
        IMessageInfo messageInfo = gp.getMessageInfo(Ui.class,
            "password.mix.case", null);
        String msg = messageInfo.getLocalizedMessage(false);
        throw new InvalidPasswordException(msg);
    }

    boolean hasNumberOrSpecialChar = Pattern.matches(".*[\\W\\d]+.*$",
candidatePassword);
    if (!hasNumberOrSpecialChar) {
        IMessageInfo messageInfo = gp.getMessageInfo(Ui.class,
            "password.special.char", null);
        String msg = messageInfo.getLocalizedMessage(false);
        throw new InvalidPasswordException(msg);
    }
}

/**
 * Check the candidate password to make sure the value satisfies the
complexity
 * requirements. If the password is not complex enough, this method
should throw
 * an {@link InvalidPasswordException} with the reason. If no exception

```

```

is thrown
 * the password is valid.
 *
 * @param user the user whose password is being checked.
 * @param candidatePassword the password to check
 * @throws InvalidPasswordException if the password is not valid
 */
@Override
public void checkPasswordForExistingUser(IDirPrincipal user,
    String candidatePassword) throws InvalidPasswordException {
    checkPasswordForNewUser(candidatePassword);
}

/**
 * Return how long a password is valid (in milliseconds) before it
 * expires. The user
 * will not be able to login after this time duration has expired
 * and the user must be
 * reset by an administrator or a custom reset password page.
 *
 * @param user the user to get the expiration value for.
 * @return return -1 for no password expiration, or the duration (in ms)
 */
@Override
public long getPasswordExpirationDuration(IDirPrincipal user) {
    try {
        if ("sysadmin".equals(user.getName())) {
            return -1; //don't expire the sysadmin password.
        }
    } catch (Exception e) {
        logger.warn(e.getMessage(), e);
    }
    return 1 * 60 * 60 * 1000; //1 hour
}

/**
 * Returns a description of the expected pattern a password must have.
 * This text
 * is displayed on the create user and update user pages of the UI.
 *
 * @return password descriptive text.
 */
@Override
public String getPasswordPatternText() {
    IMessageInfo messageInfo = gp.getMessageInfo(Ui.class,
        "password.complexity.description", null);
    return messageInfo.getLocalizedName(false);
}
}

```

2. Deploy the application to My webMethods Server.
3. Configure My webMethods Server to use your custom class on the **Properties** page for the system directory service, as described in [“Using Password Complexity Policies” on page 294](#).
4. Debug and test the Java class.

5. If required, make further changes to the Java class and repeat steps 2 and 3 until the class works as expected.

Working with Response Header Rules

My webMethods Server enables you to create and manage rules that govern the HTTP response messages that are sent after receiving and interpreting a request message.

Viewing Response Header Rules

> To view response header rules

1. As system administrator: **Administrative Folders > Administration Dashboard > User Interface > Manage Response Header Rules.**
2. Click the **View Rules** tab if it is not already selected.

Available rules are displayed in the **Rule Name** list. The list is empty if no rules are defined. The following information appears for each rule:

- The rule name. This is a clickable link that opens the **Modify Rule** where you can view the rule conditions and make changes to the rule.
- A description of the rule as entered by the rule creator.
- Whether or not the rule is currently enabled.

Creating a Response Header Rule

> To create a response header rule

1. As system administrator: **Administrative Folders > Administration Dashboard > User Interface > Manage Response Header Rules.**
2. Click the **Create Rule** tab.
3. Do the following:
 - Type a name for the rule in the **Rule Name** field.
 - Type a description of the rule in the **Description** field.
 - If you want the rule to be enabled upon creation, select the **Enabled** check box.
 - If you do not want the rule to be enabled upon creation, clear the **Enabled** check box.

4. In the **Condition** field, use the available condition element buttons to define a condition expression for the rule. The following condition elements are available:

- **Current User(s)**
- **Group/Role Membership**
- **User Attributes**
- **Request**
- **Parent Resource**
- **Current Resource Type**
- **Resource Properties**
- **Add Operator**

Click the button you want to work with and then use the resulting dialog box to select the value you want to add to the condition expression. For example:

```
portalResource isDescendant ("webm.apps.data.page.login") ||
portalResource isDescendant ("portlet.login")
```

You can also type in an expression directly, or type to modify the expression after you create it.

5. Click **Add** next to the **Result** field and specify the response header key field name and value. You can add additional key/value pairs if needed. To remove a key/value pair, select it in the list and click **Remove**.
6. Click **Create Rule**.

Modifying a Response Header Rule

> To modify a response header rule

1. As system administrator: **Administrative Folders > Administration Dashboard > User Interface > Manage Response Header Rules**.
2. Do either of the following:
 - Click the rule name in the **Rule Name** field.
 - Click  for the rule you want to work with and then click **Modify Rule**.
3. On the **Modify Rule** tab, make changes to the rule as required:

- In the **Condition** field, use the available condition element buttons to add a condition expression to the rule, or type to modify the expression directly.
4. In the **Result** field, select a response header key field name and value and do one of the following:
 - Click **Add** to specify an additional key/value pair.
 - Click **Modify** to make changes to an existing key/value pair.
 - Click **Remove** to remove a key/value pair.
 5. Click **Modify Rule**.

Copying a Response Header Rule

You can copy a rule to create a new rule in the Manage Response Header Rules page. When you copy a rule, you provide a name and description for the new rule, and the rule condition and result information is copied from the original rule into the new rule.

Note:

When you copy a rule, the setting of the **Enabled** check box in the original rule is also copied into the new rule. If the **Enabled** check box is selected in the original rule, the new copied rule will be enabled as soon as you create it.

> To copy a response header rule

1. As system administrator: **Administrative Folders > Administration Dashboard > User Interface > Manage Response Header Rules**.
2. Click  for the rule you want to work with and then click **Copy Rule**.
3. On the **Copy Rule** tab, type a name and description for the new rule.
4. Click **Copy the Rule**.

Removing a Response Header Rule

> To remove a response header rule

1. As system administrator: **Administrative Folders > Administration Dashboard > User Interface > Manage Response Header Rules**.
2. Click  for the rule you want to work with and then click **Remove Rule**.

3. Click **OK** in the confirmation dialog box to remove the rule.

Changing the Response Header Rule Evaluation Order

> To change the response header rule evaluation order

1. As system administrator: **Administrative Folders > Administration Dashboard > User Interface > Manage Response Header Rules.**
2. Click the **Change Rule Evaluation Order** tab.
3. In the **Evaluation Order** list, select a rule name and use the arrow buttons to the right to move the selected rule up or down in the evaluation order.
4. Click **Update Rule.**

Changing the Default Internet Explorer Compatibility Setting

By default, My webMethods Server sets the compatibility mode for Internet Explorer to IE8. For more information about the Internet Explorer compatibility mode in My webMethods Server, see [“About the Default Response Header Rules” on page 302.](#)

> To change the Internet Explorer document compatibility setting.

1. As system administrator: **Administrative Folders > Administration Dashboard > User Interface > Manage Response Header Rules.**
2. Locate the **IE - parameter for compatibility mode** rule and do either of the following:
 - Click the rule name in the **Rule Name** field.
 - Click  for the rule you want to work with and then click **Modify Rule.**
3. Click **Update** next to the **Result** field, edit the **Value** field, and click **Apply.**

The default value is IE-8.
4. Click **Update Rule.**

About the Default Response Header Rules

The following table lists the response header rules, available in a typical My webMethods Server installation:

Rule Name	Description
Login Page Deny Non Same-Origin Framing	<p>Enabled by default. This rule guards against cross-site scripting and clickjacking attacks on the Login page by implementing the X-Frame-Options HTTP response header. This header indicates whether or not a browser should be allowed to render a page in a <frame> or <iframe>, thus ensuring that content is not embedded into other sites. The key/value pair is:</p> <pre>X-Frame-Options SAMEORIGIN</pre> <p>The page can only be displayed in a frame of the same origin as the page itself.</p>
Login Page Deny All Framing	<p>Disabled by default. This is a more stringent Login page anti-cross-site scripting and clickjacking rule. The key/value pair is:</p> <pre>X-Frame-Options DENY</pre> <p>In this case, the page cannot be displayed in a frame, regardless of the site attempting to do so.</p>
IE - parameter for compatibility mode	<p>Enabled by default. This setting sets the standard document type for Internet Explorer in rendering HTML pages. The default value is IE8.</p>

Basic support for the X-Frame-Options header response is available in these (and later) browser versions:

- Chrome 4.1.249.1042
- Firefox 3.6.9
- Gecko 1.9.2.9
- Internet Explorer 8.0
- Opera 10.5
- Safari 4.0

Content Security Policy Settings

You can specify custom header response rules that modify the default Content-Security-Policy header for particular server resources, or use additional JVM parameters to configure the policy globally for My webMethods Server. Use the following custom JVM parameters to enable or disable the default content security policy, or configure sources of trusted content:

- `com.webmethods.content.security.disabled` - The default value is `false`. Set to `true` to disable the content security policy.
- `com.webmethods.content.security.hosts` - Use this parameter to supply additional allowed hosts. Separate multiple values with intervals.

For more information about adding JVM properties in the `custom_wrapper.conf` file for My webMethods Server, see [“Configuring JVM Settings for My webMethods Server”](#) on page 95.

For more information about working with response header rules, see [“Working with Response Header Rules”](#) on page 299.

16 Analysis, Reporting, and Troubleshooting

■ About Analysis, Reporting, and Troubleshooting	306
■ Controlling Server Logging	306
■ Viewing Logging Messages	318
■ Managing Security Audit Logging	318
■ Monitoring Real-Time User Activity	319
■ Collecting Data About Server Events	319
■ Collecting Data About Database Changes	322
■ My webMethods Server Diagnostic Tools	322

About Analysis, Reporting, and Troubleshooting

My webMethods Server provides administrators with a number of tools for analyzing, reporting, managing, and maintaining server deployment.

Controlling Server Logging

As My webMethods Server runs, it collects logging information for a variety of categories, for web applications, and for portlet applications. Logging information is collected in two stages, **Logger** and **Output**.

- **Logger** stage allows you to define the level of messages you want to collect for each category, web application, and portlet application.
- **Output** stage determines the level of messages that you want My webMethods Server to write to the console and the log files.

You can control each stage independently. The logger threshold takes precedence over an output threshold. If My webMethods Server discards a logging message because it does not meet a logger threshold, and is therefore not collected for a category or application, that message cannot be written to the output, that is, it cannot be written to the console or a log file.

Use the Logging Configuration page to control logging for the server. For more information, see [“About Logging Thresholds” on page 306](#) and [“Setting Logger and Output Thresholds” on page 308](#).

As My webMethods Server writes log messages to the output log files, the files will grow in size. Periodically, the My webMethods Server rolls over to a new set of files, making it easier to locate a specific date and to discard old log files as needed. For more information about the output log files, how often My webMethods Server rolls over files, and how you can control the rollover, see [“About the Log-File Rollover Period” on page 310](#) and [“Modifying the Log-File Rollover Period” on page 312](#).

About Logging Thresholds

Define logging thresholds to control the log messages that My webMethods Server collects. Each log message is assigned a log level. A threshold indicates the log level of messages you want My webMethods Server to collect. My webMethods Server logs messages that have the level you specify and all higher levels. As a result, by setting thresholds, you can limit the growth of log files. Set lower log levels when you want to collect more information and higher log levels when you want to collect less information.

The following table describes the levels, from the lowest level to the highest:

Log level	Description
TRACE	Set a threshold to this level if you want the logs to contain messages of all levels. This level provides the most detail; however, log files grow quickly.

Log level	Description
DEBUG	The server issues DEBUG messages at multiple points within a server event. Set a threshold to this level to collect DEBUG messages and all higher-level messages (e.g., INFO, WARN, etc.). This level is useful for debugging a problem; however, log files grow quickly.
INFO	The server issues INFO messages to indicate that a server event has occurred. Set a threshold to this level to collect INFO messages and all higher-level messages.
WARN	The server issues WARN messages to warn you of an error that is not serious. Set a threshold to this level to collect WARN, ERROR, and FATAL messages.
ERROR	The server issues ERROR messages when a non-fatal error occurs. Set a threshold to this level to collect ERROR and FATAL messages.
FATAL	The server issues ERROR messages when a fatal error occurs. Set a threshold to this level to collect only FATAL messages.

To define a threshold, assign one of the levels described in the table above.

- **Assign a log level to a logger threshold for a category, web application, or portlet application** to control the messages that My webMethods Server collects for that category or application. You can view the categories and applications for which My webMethods Server can collect messages on the Logging Configuration page.
- **Assign a log level to an output threshold** to control the messages that My webMethods Server writes to the console or one of the following log files:
 - **_full_log**, which can contain all level of messages from all categories, web applications, and portlet applications.
 - **_problems_log**, which contains messages from all categories, web applications, and portlet applications; however, it restricts the level to WARN messages or higher, that is WARN, ERROR, or FATAL messages.
 - **_errors_xm_**, which is an XML fragment that contains messages from all categories, web applications, and portlet applications. By default it contains only FATAL messages.

Note:

You can wrap the XML fragment that is contained in the errors log with a root XML element to produce well-formed XML.

The logger threshold takes precedence over an output threshold. When the server collects a log message, it first sends the message to a specified logger. If the message does not meet the threshold, it is discarded. However, if the log level of the message meets or exceeds the logger threshold, the server forwards the message on to the logging outputs. If the log level of the message does not meet the output threshold, it is discarded. Finally, if the message meets or exceeds the output threshold, the server writes the message to the output. In other words, if a message does not meet the threshold you set for a category or application (logger threshold), it is discarded and therefore cannot be written to the output.

Setting Logger and Output Thresholds

You can set both logger thresholds and output thresholds on the Logging Configuration page.

Note:

When you update the thresholds on the Logging Configuration page, the settings are permanent until you update them again. If you want My webMethods Server to temporarily use debug settings that last only until the server is shut down, you can start the server using the `-d` startup option. For more information, see [“Temporarily Setting Debug Thresholds” on page 308](#).

➤ To set a logger or output threshold

1. As system administrator, click **Administration Dashboard > Analysis > Logging Configuration**.
2. For each category, web application, and/or portlet application you want to modify, in the **Logger Threshold** column, select the log level to the lowest level of message you want to accept.
3. For each output threshold you want to modify, in the **Output Threshold** list, select the log level to the lowest level of message you want to accept.

Tip:

To set all logger thresholds or output thresholds to the same logging level, click  to the right of the **Logger Threshold** or **Output Threshold** label and then select the log level to use.

4. Click **Apply**.

Temporarily Setting Debug Thresholds

By default, the server configuration uses a java system property (`log4j.default.log.level`) to set the logging threshold for several categories to the WARN threshold. When you start the server with the `-d` startup option, the `log4j.default.log.level` variable is set to DEBUG. As a result, the server collects messages for the categories at the DEBUG threshold until the server is shut down. Use this the `-d` startup option to run the server in debug mode. This is an easy way to temporarily increase the log level for many categories to perform troubleshooting.

The following section of the logging configuration shows the categories that are affected by the `-d` startup option:

```
# default level controled by -d=DEBUG otherwise =WARN
log4j.category.Framework=${log4j.default.log.level}
log4j.category.frameworkInit=${log4j.default.log.level}
log4j.category.dataAccess=${log4j.default.log.level}
log4j.category.jsp=${log4j.default.log.level}
log4j.category.jsf=${log4j.default.log.level}
log4j.category.directory=${log4j.default.log.level}
```

```

log4j.category.portlet=${log4j.default.log.level}
log4j.category.classManager=${log4j.default.log.level}
log4j.category.taglibs=${log4j.default.log.level}
log4j.category.mail=${log4j.default.log.level}
log4j.category.search=${log4j.default.log.level}
log4j.category.messaging=${log4j.default.log.level}
log4j.category.notifications=${log4j.default.log.level}
log4j.category.schedule=${log4j.default.log.level}
log4j.category.version=${log4j.default.log.level}
log4j.category.task=${log4j.default.log.level}
log4j.category.webservice=${log4j.default.log.level}
log4j.category.wsclient=${log4j.default.log.level}

```

For instructions for how to start the server using the `-d` startup option, see [“Basic Command Line Syntax for My webMethods Server”](#) on page 72.

Exporting Threshold Settings to a File

You can export your logger and output threshold settings to a file named `log4.override.properties`. Then later, you can import them from the `log4.override.properties` file into a My webMethods Server database.

You might want to export your threshold settings if you want to save a backup copy or if you want to use the same settings in another My webMethods Server instance that is not in the same cluster.

➤ To export threshold settings

1. As system administrator, click **Administration Dashboard > Analysis > Logging Configuration**.
2. Click **Export**.
3. If you want to open the file in a text editor to view it before saving:
 - a. Select the **Open with** option.
 - b. Select the text editor you want to use to view the file.
 - c. Click **OK**.
My webMethods Server downloads the file from the database and opens it in the selected editor.
 - d. Use the text editor save function to save the file.
4. If you want to save the file without opening it:
 - a. Select **Save File**.

- b. Click **OK**.

My webMethods Server downloads the threshold settings to the `log4.override.properties` file and places the file on your desktop.

Importing Threshold Settings from a File

If you previously exported the logger and output threshold settings to the `log4.override.properties` file, you can use the following procedure to import them into a My webMethods Server instance.

> To import threshold settings

1. As system administrator, click **Administration Dashboard > Analysis > Logging Configuration**.
2. Click **Import**.
3. Click **Browse**, navigate to the location of the `log4.override.properties` file containing the settings you want to import, and click **Open**.
4. In the **Import Mode** field, select whether you to merge the settings or completely replace the settings.
 - Select **Merge with Existing Configuration** to merge the settings in the `log4.override.properties` file with the current threshold settings in the My webMethods Server database.
 - Select **Replace Existing Configuration** if you want to completely replace the threshold settings in the My webMethods Server database with the settings in the `log4.override.properties` file.
5. Click **Import Configuration**.

About the Log-File Rollover Period

Periodically, the logging process rolls over to a new set of files based on either date or file size. When My webMethods Server rolls over a file based on date, it renames the old log file so that it includes the date so that if you need to refer to old log data, it is easier to find the data for a specific date. When My webMethods Server rolls over based on size, it maintains a specified number of backup files and discards older data.

My webMethods Server log files reside in the following directory:

```
Software AG_directory \MWS\server\serverName\logs
```

You can delete server logs manually from that directory, or develop custom logic to remove logs based on the log appender timestamps and the rollover periods you configure.

The following table lists the log files that My webMethods Server creates and the default rollover period for the logs.

Log file	Default contents	Default rollover period
full.log	Log messages from all categories and all enabled thresholds (for example, TRACE, DEBUG, INFO, WARN, ERROR, and FATAL).	<p>Daily at midnight</p> <p>When the log is rolled over, the past days log messages are rolled over into a file that is named to reflect the date of the log information it contains:</p> <p>_full_.yyyy-MM-dd.log</p> <p>You can customize how often My webMethods Server rolls over this log.</p>
problems.log	WARN, ERROR, and FATAL log messages from all categories.	<p>Daily at midnight</p> <p>When the log is rolled over, the past days log messages are rolled over into a file that is named to reflect the date of the log information it contains:</p> <p>_problems_.yyyy-MM-dd.log</p> <p>You can customize how often My webMethods Server rolls over this log.</p>
install.log	DEBUG level log messages from the install	<p>When the log size reaches 100 MB</p> <p>When the log is rolled over, the old log messages are rolled over into a backup file that uses the following naming convention, where <i>N</i> is a number.</p> <p>install.<i>N</i>.log</p> <p>For example, the first time the log is rolled over, the backup log is named install.1.log. By default, My webMethods Server maintains up to three backups.</p> <p>You can customize the maximum size limit for this file before My webMethods Server rolls it over and how many backup files the server maintains.</p>
caf.log	CAF application log messages	<p>Daily at midnight</p> <p>When the log is rolled over, the past days log messages are rolled over into a file that</p>

Log file	Default contents	Default rollover period
		<p>is named to reflect the date of the log information it contains:</p> <p>caf.yyyy-MM-dd.log</p> <p>You <i>cannot</i> customize how often My webMethods Server rolls over this log.</p>
schema.log	Log messages from DDL statements execution for creating, modifying, or deleting x-type objects	<p>When the log size reaches 5 MB</p> <p>When the log is rolled over, the old log messages are rolled over into a backup file that uses the following naming convention, where <i>N</i> is a number.</p> <p>schema.<i>N</i>.log</p> <p>For example, the first time the log is rolled over, the backup log is named schema.1.log. By default, My webMethods Server maintains up to ten backups.</p> <p>You can customize the maximum size limit for this file before My webMethods Server rolls it over and how many backup files the server maintains.</p>
errors.xm_	<p>An XML fragment of FATAL log messages from all categories</p> <p>You can wrap the XML fragment that is contained in the errors log with a root XML element to produce well-formed XML.</p>	This log is <i>not</i> rolled over

For information about how to customize the rollover periods, see [“Modifying the Log-File Rollover Period” on page 312](#).

Modifying the Log-File Rollover Period

My webMethods Server maintains the settings for how to roll over the log files in the My webMethods Server database. As a result, if you are running the server in the cluster, all server instances use the same rollover settings. To update the rollover settings, you can have My webMethods Server download the settings to a local file named log4j.init.properties file, modify the rollover periods in the downloaded file, and then upload the changes back to the database.

If My webMethods Server rolls over the log file based on:

- **Date and time, (used for the `_full_.log` and `_problems_.log` files)**

The rollover configuration is controlled by the appender type `org.apache.log4j.DailyRollingFileAppender`. To update the rollover period, you identify a new `DatePattern` for the appender. The following table describes the `DatePatterns` you can use to configure log rollover:

Date Pattern	Rollover Time
' . 'yyyy-MM	<p>At the beginning of each month</p> <p>For example, for the <code>_full_.log</code> if you set the <code>DatePattern</code> to <code>' . 'yyyy-MM</code>, at midnight on January 31, 2010, My webMethods Server copies the log data to the file <code>_full_.2010-01.log</code>. My webMethods Server logs messages for the month of February to the <code>_full_.log</code> file until it is rolled over the next month.</p>
' . 'yyyy-ww	<p>At the first day of each week. The first day of the week depends on the locale.</p> <p>For example, assume the first day of the week is Sunday and that for the <code>_problems_.log</code> you set the <code>DatePattern</code> to <code>' . 'yyyy-ww</code>. On Saturday midnight, May 15, 2010, My webMethods Server copies the log data for the 19th week of the year to the file <code>_problems_.2010-19</code>. My webMethods Server logs messages for the 20th week of 2010 to the <code>_problems_.log</code> file until it is also rolled over the next week.</p>
' . 'yyyy-MM-dd	<p>At midnight each day.</p> <p>For example, for the <code>_full_.log</code> if you set the <code>DatePattern</code> to <code>' . 'yyyy-MM-dd</code>, at midnight on February 22, 2010, My webMethods Server copies the log data to the file <code>_full_.2010-02-22</code>. My webMethods Server logs messages for February 23, 2010 to the <code>_full_.log</code> file until it is also rolled over the next day.</p>
' . 'yyyy-MM-dd-a	<p>Twice each day, at noon and midnight.</p> <p>For example, for the <code>_problems_.log</code> if you set the <code>DatePattern</code> to <code>' . 'yyyy-MM-dd-a</code>, at noon on February 22, 2010, My webMethods Server copies the log data to the file <code>_problems_.2010-02-22-A.M.</code> My webMethods Server logs messages for the afternoon of February 22, 2010 to the <code>_problems_.log</code> file until it is rolled over at midnight into the file <code>_problems_.2010-02-22-P.M.</code> Then, My webMethods Server logs messages for February 23, 2010 to the <code>_problems_.log</code> file.</p>
' . 'yyyy-MM-dd-HH	<p>Every hour of each day.</p> <p>For example, for the <code>_full_.log</code> if you set the <code>DatePattern</code> to <code>' . 'yyyy-MM-dd-HH</code>, at approximately 11:00 A.M. on February 22, 2010, My webMethods Server copies the log data for the 10 o'clock hour to the file <code>_full_.2010-02-22-10</code>. My webMethods Server logs</p>

Date Pattern	Rollover Time
	messages for the 11 o'clock hour to the <code>_full_.log</code> file until it is rolled over at the beginning of the next hour.
' . 'yyyy-MM-dd-HH-mm	Every minute of each day. For example, for the <code>_full_.log</code> if you set the <code>DatePattern</code> to ' . 'yyyy-MM-dd-HH-mm, at approximately 11:46 A.M. on February 22, 2010, My webMethods Server copies the log data to the file <code>_full_2010-02-22-11-45</code> . My webMethods Server logs messages for the next minute to the <code>_full_.log</code> file until it is rolled over a minute later.

■ File size, (used for the `install.log` file)

The rollover configuration is controlled by the `org.apache.log4j.RollingFileAppender`. You can update the log level of the messages collected in the `install.log` file, the maximum size of the log file before it is rolled over, and the number of backup log files that My webMethods Server maintains.

Note:

The following procedure describes how to modify the roll over periods by updating parameters for the default appenders (`org.apache.log4j.DailyRollingFileAppender` and `org.apache.log4j.RollingFileAppender`) that are defined out-of-the-box. However, if you want, you can update the settings to use an alternative appender to meet your needs. For example, you can change the `_problems_.log` to use the `RollingFileAppender` if you want it to roll over based on size or `FileAppender` if you do not want the log to roll over. You can use any appender that `log4j` library supports. For more information, see <http://logging.apache.org/log4j/1.2/index.html>.

➤ To modify the log-file rollover period

1. Download the `log4j.init.properties` file from the My webMethods Server database:

a. At a command line prompt, change directories to move to the server's bin directory:

```
Software AG_directory \MWS\bin
```

b. To retrieve the `log4j.init.properties` file from the database, type this command:

```
mws -s serverName getConfig log4j.init.properties
```

2. Edit the `log4j.init.properties` file.

a. Navigate to the following directory, where the `getConfig` command placed the `log4j.init.properties` file:

```
Software AG_directory \MWS\server\serverName\config
```

- b. Open the `log4j.init.properties` file in a text editor.
3. To modify the rollover settings for the `_full_.log` file:
 - a. Locate the following portion of the file:

```
log4j.appender.rootFile=org.apache.log4j.DailyRollingFileAppender
```

```
log4j.appender.rootFile.DatePattern='.'yyyy-MM-dd
```

```
log4j.appender.rootFile.File=${log4j.logging.dir}/_full_.log
```

- b. Update the date pattern to define when you want the `_full_.log` file to rollover.
4. To modify the rollover settings for the `_problems_.log` file:
 - a. Locate the following portion of the file:

```
log4j.appender.rootErrorsFile=org.apache.log4j.  
DailyRollingFileAppender
```

```
log4j.appender.rootErrorsFile.DatePattern='.'yyyy-MM-dd
```

```
log4j.appender.rootErrorsFile.File=${log4j.logging.dir}/  
_problems_.log
```

- b. Update the date pattern to define when you want the `_problems_.log` file to rollover.
5. To modify the rollover settings for the `install.log` file:
 - a. Locate the following portion of the file:

```
log4j.appender.installFile=org.apache.log4j.RollingFileAppender
```

```
log4j.appender.installFile.threshold=DEBUG
```

```
log4j.appender.installFile.MaxFileSize=100MB
```

```
log4j.appender.installFile.MaxBackupIndex=3
```

```
log4j.appender.installFile.File=${log4j.logging.dir}/install.log
```

- b. To change the size limit of the file, update the value on the `log4j.appender.installFile.MaxFileSize` line to specify an alternative maximum file size.
- c. To change the number of backups that you want My webMethods Server to maintain, update the value on the `log4j.appender.installFile.MaxBackupIndex` line.

It is recommended that you do not update the `log4j.appender.installFile.threshold` line to change the threshold level for the `install.log`. Because the installation process issues INFO messages, if you raise the threshold, no messages will be logged.

6. To modify the rollover settings for the schema.log file:

a. Locate the following portion of the file:

```
log4j.appender.DDLSchemaFileAppender=org.apache.log4j.  
RollingFileAppender  
  
log4j.appender.DDLSchemaFileAppender.threshold=INFO  
  
log4j.appender.DDLSchemaFileAppender.MaxFileSize=5MB  
  
log4j.appender.DDLSchemaFileAppender.MaxBackupIndex=10  
  
log4j.appender.DDLSchemaFileAppender.File=${log4j.logging.dir}  
/schema.log  
  
log4j.appender.DDLSchemaFileAppender.layout=org.apache.log4j.  
PatternLayout  
  
log4j.appender.DDLSchemaFileAppender.layout.ConversionPattern=  
${log4j.mes  
sage.pattern}
```

- b. To change the size limit of the file, update the value on the `log4j.appender.DDLSchemaFileAppender.MaxFileSize` line to specify an alternative maximum file size.
- c. To change the number of backups that you want My webMethods Server to maintain, update the value on the `log4j.appender.DDLSchemaFileAppender.MaxBackupIndex` line.

7. Save the `log4j.init.properties` file.

8. Deploy the revised file to the My webMethods Server database:

- a. At a command line prompt, change directories to move to the server's bin directory:

```
Software AG_directory \MWS\bin
```

- b. To write the
- `log4j.init.properties`
- file back to the database, type this command:

```
mws -s serverName putconfig log4j.init.properties
```

- c. Delete the local copy of the
- `log4j.init.properties`
- file.

If you do not delete the file, the server instance will continue to use the local version of the file.

9. Restart the cluster.

Changes do not take effect until the cluster is restarted.

Changing the Default Logging Directory

By default, logs for a server instance reside in this location:

Software AG_directory \MWS\server\serverName\logs

You can change the location of this directory by modifying the `systemPaths.properties` file.

➤ Changing the default logging directory

1. Download the `systemPaths.properties` file from the My webMethods Server database:

- a. At a command line prompt, change directories to move to the server's bin directory:

```
Software AG_directory \MWS\bin
```

- b. To retrieve the `systemPaths.properties` file from the database, type this command:

```
mws -s serverName getconfig systemPaths.properties
```

2. Open, in a text editor, the `systemPaths.properties` file, which you will find in this location:

Software AG_directory \MWS\server\serverName\config

3. Modify this line to point to the new location of the logs directory, and save the file:

```
system.path.logs=root:/logs
```

4. Deploy the revised file to the My webMethods Server database:

- a. At a command line prompt, change directories to move to the server's bin directory:

```
Software AG_directory \MWS\bin
```

- b. To write the `systemPaths.properties` file back to the database, type this command:

```
mws -s serverName putconfig systemPaths.properties
```

- c. Delete the local copy of the `systemPaths.properties` file.

If you do not delete the file, the server instance will continue to use the local version of the file.

5. Restart the server instance.

Changes do not take effect until the server is restarted.

Viewing Logging Messages

The Log Viewer page allows you to view the latest messages in the `_full_.log`, `_problems_log`, and `_errors_.xm` files.

For the Log Viewer page to be useful, you need to make sure you are collecting the right messages for your needs. Use the Logging Configuration page to set message collection criteria. For more information, see [“About Logging Thresholds” on page 306](#) and [“Setting Logger and Output Thresholds” on page 308](#).

> To view logging messages

1. As system administrator, click **Administration Dashboard > Analysis > Log Viewer**.
2. Select the number of lines that you want to view.
3. Select from which log you want to view messages.
4. Click **Refresh Now**.

Managing Security Audit Logging

By default, My webMethods Server performs audit logging on all auditable events. Using the Audit Administration page, you can enable or disable audit logging, or choose which events are to be logged. By default, audit logging is enabled and all available auditable events are logged.

My webMethods Server writes the audit log to the `audit.log` file, which resides in the following directory:

Software AG_directory \MWS\server\serverName\logs

> To manage security audit logging

1. As system administrator, click **Administration Dashboard > Configuration > Audit Administration**.
2. To enable or disable audit logging:
 - a. To enable audit logging, select **Enable Auditing**.
This is the default setting.
 - b. To disable audit logging, clear **Enable Auditing**.
3. In the Auditable column, select events to be logged and clear events that are not to be logged.

Selected events are not logged if audit logging is disabled.

4. Click **Apply**.

The changes take effect immediately.

Monitoring Real-Time User Activity

The Session Monitor page can be used to monitor real-time user activity for a server deployment and send status messages to active users by means of E-mail. For active users, a system administrator can accomplish two important functions:

- view a user's profile information
- send the user E-mail directly from within this page

> To view all active server sessions

1. As system administrator, click **Administration Dashboard > Analysis > Session Monitor**.
2. (Optional) On the list of active sessions, click a user's name to view that user's profile.
3. (Optional) On the list of active sessions, click .

If you have an E-mail client installed on the machine you are working on, an E-mail message window is displayed allowing you to compose and send an E-mail to the selected user. If the user does not have a valid E-mail address in User Information, the **To** field is empty.

Collecting Data About Server Events

The Events Collector page collects data about events on the server so they can be used by other pages.

When deployed on a server, the Events Collector page captures information about certain types of server events and places them in the server database. The page can capture the following events:

- Login and logout.
- Get events, such as when a user browses a page.
- Operation events, such as when an object is created, updated, moved, or deleted.

For each event, the page collects information on the user associated with the event, the date and time of the event, the host name of the machine, and where possible, information about the operation being performed.

To take advantage of data collected by this page, another page performs a query against the server database, and then displays the results on a page. You can find examples of pages that perform these queries on the Software AG TECHcommunity website at <http://>

techcommunity.softwareag.com. To try one or more of these pages, you need to take the following actions:

1. Deploy the Events Collector page and the sample pages on the server.
2. Configure the Events Collector page.
3. Populate a page with the sample portlets that display server events.

Sample pages include portlet source code so you can import the portlets into Software AG Designer and see how they function. For an example of the database schema used by the Events Collector page for placing data into the server database, see [“Events Collector Database Schema” on page 321](#).

Deploying the Events Collector Page

The Events Collector page is part of a standard My webMethods Server installation but is not deployed by default. Before you can use the page, you must first deploy it on the server.

> To deploy the Events Collector Configuration page on a server

1. Locate the Events Collector Configuration page at this location in the My webMethods Server directory structure:

```
Software AG_directory \MWS\components\extras\analysis\wm_eventscollector.pdp
```

2. Copy the `wm_eventscollector.pdp` file and paste it into the Deploy directory:

```
Software AG_directory \MWS\server\server_name\deploy
```

where `server_name` is the name of the server. After several seconds, the page is automatically deployed on the server.

Configuring the Events Collector Configuration Page

By default, the Events Collector Configuration page is ready to begin collecting data on events as soon as you deploy it, but you may want to change how long data is kept, or to disable the page.

> To configure the Events Collector Configuration page

1. As system administrator, click **Administration Dashboard > Analysis > Events Collector Configuration**.

Note:

If you cannot find the Events Collector Configuration page in the Analysis folder, it may not be deployed. For more information, see [“Deploying the Events Collector Page” on page 320](#).

2. Check or clear the **Collection Enabled** check box to enable or disable the collection of server event data.

By default, data collection is enabled.

3. In the **Keep Data for** list, select how long to keep server event data.

Choices range from **One Day** to **One Year**. The default is **One Month**.

4. Click **Apply**.

Events Collector Database Schema

The Events Collector page uses the following database schema.

```
tblwEvents (main table where events data is being collected)
  idEvent      - Primary key
  idType       - Foreign key to tblwEventTypes. Stores the type
                of an event.
  idHost       - Foreign key to tblwEventHosts. Stores the host
                where the event occurs.
  timestamp    - Time stamp of an event, defined as the number of
                milliseconds since epoch(java.lang.System.
                currentDateMillis()).
  userID       - Database ID of the user who performed an operation.
  thingID_1    - object_1. For example, for Get type events this is
                the database ID of the object being viewed.
  thingID_2    - object_2. Used in rare cases where two objects are
                involved, for example when an object is created.
                Then object_1 is the database ID of the container,
                and object_2 is the database ID of an object that
                was created.
  action       - Used for Login events: 1 - user logged in, 2 - user
                logged out.
tblwEventHosts (stores mapping between hostID and hostname)
  idHost       - Host id.
  hostname     - Actual host name where the event occurs.
tblwEventTypes (stores mapping between eventID and eventTypeName)
  idType       - Event type ID.
  eventType    - Event type name.
Possible eventType name values:
  com.webmethods.portal.event.add.impl.CreateEvent
    - New object is created.
  com.webmethods.portal.event.impl.GetEvent
    - Object is being browsed.
  com.webmethods.portal.event.system.impl.LoginEvent
    - User logs in/out.
  com.webmethods.portal.event.modify.impl.UpdateEvent
    - Object is updated.
  com.webmethods.portal.event.remove.impl.DeleteEvent
    - Object is deleted.
```

Collecting Data About Database Changes

You can collect logging information about DDL statements execution to the My webMethods Server database. All data about database changes is collected and written to the `schema.log` file. You can find the setting for logging database changes information on the Administration Dashboard > Analysis > Logging Configuration page.

For more information on setting up the logging thresholds, see [“Controlling Server Logging” on page 306](#).

For more information on how to view the collected data, see [“Viewing Logging Messages” on page 318](#).

My webMethods Server Diagnostic Tools

You can use the My webMethods Server diagnostic tools to capture and analyze data about server operation. The My webMethods Server diagnostic tools have two types of tools, diagnostic command line tools and diagnostic portlets. This topic contains a brief overview of the diagnostic tools. For complete information, see *Diagnosing My webMethods Server*.

With the diagnostic command line tools, you can do the following:

- Use the `threaddump` tool to capture data on thread execution for servers on a local or remote system
- Use the `envcapture` tool to capture environment-specific server information to be provided to Software AG Global Support for troubleshooting assistance
- Use the `envdiff` tool to compare XML files that result from the capture of environment-specific server information
- Use the `memorydump` tool to capture server memory allocation information
- Use the `loganalyzer` tool to identify and analyze issues reported in the `errors.log` file

With the diagnostic portlets you can do the following:

- Use the `thread Dump` tool to monitor thread execution deadlocks in server threads
- Use the `Performance Analysis` tool to measure and analyze the performance of services and custom portlets or applications using an embedded performance-monitoring service
- Use the `Performance Statistics` tools to display statistic and analysis information about the performance of various server actions, which are grouped by categories
- Use the `Log Analysis` tools to read and analyze log files created within the server in accordance with the `log4j` mechanism
- Use the `Memory Monitor` tool to monitor the memory usage of the Java Virtual Machine (JVM) and send notifications to administrators when the configured threshold limits are reached

17 My webMethods Server Configuration

■ About My webMethods Server Configuration	324
■ Managing Aliases	324
■ Deploying My webMethods Server Components	328
■ Configuring My webMethods Server Cache	331
■ Displaying System Information	337

About My webMethods Server Configuration

My webMethods Server provides administrators with a number of tools that can be used to help configure your server. You perform these functions after installing and configuring a My webMethods Server instance.

Managing Aliases

The Alias Management page lets you manage URL aliases as server objects. With this page, you can create, view, modify, or delete custom URL aliases and create more friendly URLs for various parts of your server.

For example, if you want to create an area of the server for the Sales Department, and you have already created a folder for the Sales team in your server's Public Folders, it might be referenced by a non-intuitive URL such as: `http://server:port/meta/default/folder/0000002216`

To make it easier for the Sales team to remember the location of the Sales server, you can use the Alias Management page to create a more user friendly URL such as: `http://server:port/Sales`

Creating an Alias to a Server Resource on the Properties Page

You can create an alias for a server resource on the Properties page of the server resource:

➤ To create an alias to a server resource on the Properties page

1. As system administrator, navigate to the page where the server resource is located.
2. On the server resource, click  and then click **Properties**.
3. On the **Alias** field of the Properties page, click **Add**.
4. Type an alias name for the server resource and click **OK**.

Do not include spaces in your alias name, or the alias will not function properly.

5. Click **Apply**.

Creating an Alias to a Server Resource on the Alias Management Page

You can create an alias for a server resource on the Alias Management page:

➤ To create an alias to a server resource on the Alias Management page

1. As system administrator, click **Administration Dashboard > Configuration > Alias Management**.
2. Click **Create**.
3. In the **Alias Name** field, type the name for the new alias you want to create, for example Sales.
Do not include spaces in the alias name.
4. For **Target**, select one of the following targets for the new alias:
 - **Resource** - targets a server resource, such as a folder, a portlet or an item. To specify the resource, click **Browse** and move the resource to the **Selected Items** box, then click **Select**.
 - **Path** - targets an external resource, such as an URL. Specify the path to the external resource, for example `http://www.softwareag.com`.

Note:

When using server resources for alias targets, you can use the **Append this string** field to pass parameters or invoke a server command on the resource that the alias references.

5. Click **Add Alias**.
6. Confirm that your alias behaves as expected by browsing to the user-friendly URL for your new alias.

By default, when you create an alias, it is appended to the root URL for your server. For example, if you create an alias called Sales, you can access the new alias by typing the URL `http://server:port/Sales`.

Searching for Aliases

You can use the Alias Management page to search for existing aliases. The page places search results in a list from which you can modify or delete aliases, or view target resources.

Tip:

In the alias search field, you can use a single wildcard character (*) to substitute for text anywhere within the name.

Performing a Simple Alias Search

To search for aliases, use the following procedure.

➤ To perform a simple search for aliases

1. As system administrator, click **Administration Dashboard > Configuration > Alias Management > Search**.

2. Type the name of the alias you want to find.

To specifically include or exclude system aliases from search criteria, see “Specifically Including or Excluding System Aliases” on page 326.

3. Click **Go**.

All aliases that match the search appear in a table.

Specifically Including or Excluding System Aliases

To include or exclude system aliases during a search, use the following procedure.

> To include or exclude system aliases during a search

1. As system administrator, click **Administration Dashboard > Configuration > Alias Management > Search**.
2. Type the name of the alias you want to find.
3. Click **Refine**.
4. In the **Include System Aliases** list, choose one of the following:
 - **Yes** - to include system aliases in the search
 - **No** - to exclude system aliases from the search
5. Click **Go**.
6. To close the refined search panel, click **Close**.

Performing an Advanced Alias Search

To perform an advanced search for aliases, use the following procedure.

> To perform an advanced search for aliases

1. As system administrator, click **Administration Dashboard > Configuration > Alias Management > Advanced**.
2. Type the name of the alias you want to find.
3. For **Include System Aliases**, select one of the following options:
 - **Yes** - Default. Includes system aliases in the search.

- **No** - Excludes system aliases from the search.
4. For **Alias Target**, do one of the following:
 - Click **Browse**, and then move the target resource to the **Selected Items** box and click **Select**.
 - Click **Use Alias**, and then type the alias of the target resource in the **Alias Name** field. Click **Test** to determine if the alias is valid and the alias target is the correct one. If the alias is correct, click **Select**.
 5. Click **Go**.

Using Saved Alias Searches

You can save an alias search for regular use. Using saved alias searches is similar to using saved searches for users, groups, or roles, except they are performed from this location:

- As system administrator, click **Administration Dashboard > Configuration > Alias Management**.

For more information about saved searches for users, groups, or roles, see [“Working with Saved Searches” on page 144](#).

Modifying an Alias to Point to a Different Server Resource

To modify an alias to point to a different server resource, follow these steps:

➤ To modify an existing alias to point to a different server resource

1. As system administrator, click **Administration Dashboard > Configuration > Alias Management**.
2. Use the **Alias Search** panel to find the alias you want to modify.
3. In the search results, click  for the alias to be modified.
4. On the **New Target** panel, select one of the following targets for the new alias:
 - **Resource** - targets a server resource, such as a folder, a portlet or an item. To specify the resource, click **Browse** and move the resource to the **Selected Items** box, then click **Select**.
 - **Path** - targets an external resource, such as an URL. Specify the path to the external resource, for example `http://www.softwareag.com`.

Note:

When using server resources for alias targets, you can use the **Append this string** field to pass parameters or invoke a server command on the resource that the alias references.

5. Click **Update**.

Deleting an Alias

To delete an alias, use the following procedure.

➤ To delete an existing alias

1. As system administrator, click **Administration Dashboard > Configuration > Alias Management**.
2. Use the **Alias Search** panel to find the alias you want to modify.
3. In the search results, click  for the alias to be deleted.
4. In the search results, select the check boxes beside the alias you want to delete, and click **Delete**.

Deploying My webMethods Server Components

System administrators have the following options available to them when installing server components, such as portlets or DBOs, on a server.

- Through the Install Administration page on the Administration Dashboard.
- Through the Deploy folder on the server's File System. This folder allows system administrators and developers to copy or paste a newly developed portlet package (such as a portlet, CAF application, Task application, or deployable package) into a specific directory that is periodically polled by the server. If the server detects new deployable components in this folder, these components are automatically retrieved and installed on the server. You have the option to configure the polling interval that specifies how often the server will poll the Deploy directory to detect any new components.

Modifying the Polling Interval

If your organization is developing multiple portlets, this installation method may be more convenient than manually installing portlets one at a time. The default file system location for the Deploy folder is:

Software AG_directory \MWS\server\serverName\deploy

Note:

Polling can be turned on or off by modifying the PhaseProvider.xml configuration file on the server's file system. Use the following instructions to modify the polling interval.

➤ To modify the polling interval

1. At a command line prompt, type the following command to move to the server's bin directory:

```
cd Software AG_directory\MWS\bin
```

2. Retrieve the phaseProvider.xml file from the My webMethods Server database, using the getConfig command, as follows:

```
mws getConfig phaseProvider.xml
```

3. Navigate to the following location:

```
Software AG_directory \MWS\server\serverName\config
```

4. Open the phaseProvider.xml configuration file in a text editor or equivalent XML editing facility. Locate the following XML fragment:

```
<Phase name="deploySync" enabled="true"  
  class="com.webmethods.portal.system.init.impl.MasterServerPhase">
```

```
<PhaseInfo name="startTimedSyncDeploy" enabled="true"  
  class="com.webmethods.portal.bizPolicy.biz.install.impl.
```

```
SyncDeployService" interval="5" />
```

```
</Phase>
```

5. To turn polling off, change the enabled attribute from true to false.
6. To change the polling interval, modify the interval attribute to the desired value. The default setting is 5 seconds.

Note:

This setting will *not* have an impact on overall performance

7. Save the file.
8. Deploy the revised file to the My webMethods Server database, using the putconfig command, as follows:

```
mws putconfig phaseProvider.xml
```

9. Delete the file from the \serverName\config directory.

If you do not delete the file, this server instance will continue to use the local version of the configuration file.

10. Restart your server instance.

Installing a Portlet Using the Deploy Folder

To use the Deploy folder to install a portlet on a server, do the following.

> To install a portlet using the Deploy folder

- Copy and paste the server component(s) that you want to deploy into the deploy directory.

Note:

If any server component fails to deploy, the server will automatically create a Failed directory on the server's file system in the Deploy folder. All components that do not install properly will be copied into the Failed directory.

Installing Portlets or Other Deployable Server Components

To use the Install Administration page to install portlets or other deployable server components, do the following

> To install a portlet or other deployable server component

1. As system administrator, click **Administration Dashboard > Configuration > Install Administration**.
2. Click **Install New Component**.
3. Choose **Local or Network Location** or **Remote Location**.
4. Do either of the following:
 - If the deployable component resides on your local file system, click **Browse** and navigate to the location.
 - If the deployable component is in a remote location, type the complete path to the component.
5. Click **Next**.
6. Review the Component Info Summary and then click **Install**.

If the component is installed successfully, you will get a confirmation message verifying that the install succeeded.

Note:

If a component install fails, that component is automatically uninstalled. Be sure to check your log files to troubleshoot the installation failure.

Uninstalling Portlets or Other Deployable Server Components

Before you uninstall a component, determine how its removal will affect all of its instances on a user's page. Uninstalling will break any page that contains specific portlet instances of the portlet that was uninstalled, and disrupt any portlets that may be wired to that portlet using the portlet wiring feature.

For example, you are not warned about wiring relationships when removing a portlet that is wired to another portlet.

You may want to change the portlet's status property to Hidden or Disabled to phase out the portlet before you uninstall it. After users are informed of the impending uninstall and have removed it from their page, it will then be safe to uninstall it.

Important:

When an uninstalled portlet's instances are broken, it causes errors on each page on which that portlet is being used. It also may remove the data for a portlet and its instances, the configuration files, the portlet database tables, and the portlet packaging files. Reinstalling will *not* restore the broken references caused by uninstalling a portlet.

➤ To uninstall a component using the Install Administration page

1. As system administrator, click **Administration Dashboard > Configuration > Install Administration**.
2. On the tree list, select one or more components to be uninstalled, click **Uninstall Selected**, and click **Next**.
3. To confirm the uninstall action, click **Uninstall**.

Configuring My webMethods Server Cache

You modify the default cache settings for My webMethods Server components using the **Cache Configuration** portlet. The **Cache Configuration** portlet is not installed by default. For more information about how to install My webMethods Server portlets, see [“Installing Portlets or Other Deployable Server Components” on page 330](#).

For information about how to configure cache settings with the cache.xml file, see [“Modifying Cache Settings in the cache.xml File” on page 333](#)

For information about how to configure cache settings for individual OSGi components, see [“Modifying Cache Settings in the cache.xml File” on page 333](#)

➤ To configure My webMethods Server cache

1. As sysadmin, navigate to **Folders > Administrative Folders > Administration Dashboard > Configuration > Cache Configuration**.

2. If the cache type you want to configure is not listed on the first page, click **Next**.
3. In the **Cache Max Size** field, enter the new maximum cache size for the cache type that you want to configure.
4. Optionally, review the **Cache Detailed Report**.

The **Cache Detailed Report** displays detailed information about each cache entry. For more information about cache types and cache attributes, see [“Cache Types and Cache Attributes” on page 332](#).

5. Click **Apply**.

Cache settings, configured on one instance are automatically propagated across the My webMethods Server cluster.

Cache Types and Cache Attributes

Cache Types

The following table lists the cache types, available on the **Cache Configuration** portlet. Some cache types are read-only.

Cache Name	Description
AccessCache	Non-transient cache for storing the access permissions of a user.
AliasCache	Non-transient cache for storing alias keys.
BpmProcessDefinitionCache	Transient cache for storing BPM process definitions.
BpmProcessInstanceCache	Transient cache for storing BPM process instances.
ContainerCache	Transient cache for refreshing xtype containers.
GroupCache	Non-transient cache for storing the groups of which a principal is a member.
InvalidLoginsCache	Non-transient cache for storing invalid login attempt data
JMSTransactionCache	Non-transient cache for storing JMS transaction data.
PortletTransientData	Non-transient cache for storing objects, used by portlets.
PresentationCache	Non-transient cache for storing rendered HTML fragments.
RoleCache	Non-transient cache for storing the roles of which a principal is a member.
TaskDataCache	Transient cache for storing task data.

Cache Name	Description
ThingCache	Transient cache for storing the ThingIDs of My webMethods Server objects.
ThingRelationCache	Transient cache for storing the relations between ThingIDs and objects.
Transient cache for AliasCache	Transient cache for storing the access permissions of a user.
Transient cache for PresentationCache	Transient cache for storing alias keys.

Cache Attributes

The following table lists the cache attributes, available in the detailed cache report for cache types on the **Cache Configuration** portlet. Some attributes are specific to cache types.

Cache Attribute	Description
Cache size	The number of items stored in the cache.
Maximum size	The maximum number of items to keep in the cache.
Valid entries	The number of valid entries, stored in the cache. Valid entries are items that resolve to objects.
Invalid entries	The number of invalid entries, stored in the cache. Invalid entries are items that do not resolve to objects.
Expired entries	The number of items that have reached the expiration time.
Expirable entries	The number of items that have an expiration time.
Cache hits	The number of items found in the cache.
Cache misses	The number of items not found in the cache.
Total dependencies	The total number of cache entries that have dependencies.
Entries with dependencies	The number of entries that have dependencies to other items in the cache.

Modifying Cache Settings in the cache.xml File

You use the cache.xml file for detailed configuration of My webMethods Server cache types. The cache.xml file is stored in the My webMethods Server database. Changes to the cache.xml file apply to all nodes in a My webMethods Server cluster.

The default storage location for a local copy of the file is *Software AG_directory* \MWS\server*serverName*\config. To configure a My webMethods Server instance to use a copy

of the `cache.xml` file, stored in a non-default location, add the following JVM property in the `custom_wrapper.conf` file for the instance:

```
-Dcom.webmethods.mws.cachemanager.configfile=path
```

where *path* is the path to the directory that contains the `cache.xml` file. For more information about configuring JVM properties, see [“Configuring JVM Settings for My webMethods Server” on page 95](#).

Unless you specify custom settings in the `cache.xml` file, cache settings are applied using the OSGi properties of each cache type, stored in the `OSGI-INF/` directory of the component jar for the type. You can configure cache settings individually for each component, as described in [“Modifying Cache Settings with Custom Property Files” on page 335](#).

➤ To modify cache settings in the `cache.xml`

1. In a command line prompt, navigate to the `bin` directory of the server:

```
Software AG_directory\MWS\bin
```

2. Retrieve the `cache.xml` file from the My webMethods Server database using the `getConfig` command:

```
mws getConfig cache.xml
```

3. Open the `cache.xml` file in a text editor and edit cache settings as required.

The default download location of the `cache.xml` file is `Software AG_directory\MWS\server\serverName\config`.

4. Save and close the file.
5. To deploy the updated file to the My webMethods Server database, type the following command:

```
mws putconfig cache.xml
```

6. Delete the file local copy of the `cache.xml` file from the `serverName\config` directory.

If you do not delete the copy, this server instance will continue to use the local version of the configuration file.

7. Restart the node using the `restart` command:

```
mws -s serverName restart
```

Changes to configuration files apply after restarting the server.

Modifying Cache Settings with Custom Property Files

You can create a .property file and configure cache setting individually for the OSGi components of each cache type that My webMethods Server supports. Settings, specified in a property file apply to a single cache type, and only on the current node. Settings, specified in the cache.xml file for My webMethods Server override individual cache configurations in .property files.

➤ To modify cache settings using property files

1. In a text editor, create a .properties file with the same name as the fully-qualified name of the cache class.
2. Add the required properties in key=value pairs, each pair on a separate row, without quotes and delimiters.
3. Save the properties file in the following location:

```
SoftwareAG_directory\profiles\MWS_instanceName\configuration\com.softwareag.platform.config.propsloader
```

Example

To configure the alias cache for My webMethods Server:

1. Create a file with name `com.webmethods.portal.service.cache.impl.AliasCache.properties`.
2. Set the maximum size of the of the alias cache to 1002222 entries by adding the following entry in the file: `maxSize=1002222`
3. Save the `com.webmethods.portal.service.cache.impl.AliasCache.properties` file in the `\profiles\MWS_default\configuration\com.softwareag.platform.config.propsloader` directory.

Configuring Role or Group Cache Lifecycle Calculation

By default, the role cache or group cache for a user is recalculated on every user login. Conditions in which the calculation involves user or group searches that take a long time can result in poor performance in My webMethods Server and the LDAP servers to which it is connected. A configuration option in the cache.xml file enables the role cache or group cache to be persisted across user logins, causing large performance improvements.

Because the roles are no longer calculated at user login, this optimization modifies the dynamic role capability. With the default setting, where the roles are calculated at user login, the user gains or loses role changes at that time. With the new setting, if a dynamic query role changes the results, the user does not immediately get the change. Role changes made by the system administrator will, however, be recognized. In this mode, static roles function correctly with the new setting. Dynamic roles do not return expected changes until an administrator changes role membership or adds a description, or until the various caches time out.

Under optimization, because My webMethods Server does not control group definition, the Group cache is not updated until the cache entry times out or a system restart occurs. If users change group membership frequently, group cache optimization may not be desirable. The choice of whether to perform role cache or group cache recalculation at login is a trade-off between performance and functionality.

For more information about modifying the cache.xml file for My webMethods Server, see [“Modifying Cache Settings in the cache.xml File” on page 333](#)

➤ To configure role cache or group cache lifecycle calculation

1. Retrieve the cache.xml file from the My webMethods Server database.
2. Open the cache.xml file in a text editor and locate the cache.

■ Role cache:

```
<cache name="RoleCache"
class="com.webmethods.portal.service.cache.impl.RoleCache"
  maxSize="1000"
  defaultCacheTimeout="3600000"
  roleCacheLifecycle="0"
  isClustered="true"
  ID="11"
  enabled="true"/>
```

■ Group cache:

```
<cache name="GroupCache"
class="com.webmethods.portal.service.cache.impl.GroupCache"
  maxSize="1000"
  defaultCacheTimeout="3600000"
  groupCacheLifecycle="0"
  isClustered="true"
  ID="19"
  enabled="true"/>
```

3. Modify the value of `roleCacheLifecycle` or `groupCacheLifecycle` as required:

The following table lists the possible values for group and role cache settings:

Value	Action
0	(Default) Causes the cache to be recalculated at user login.
1	Causes the cache to be persisted across user logins.

4. Save and close the cache.xml file.
5. Deploy the revised file to the My webMethods Server database, and delete the local copy of the file.

- Restart the My webMethods Server node.

Displaying System Information

The System Information page provides a wealth of information about the current state of the server. The page gathers the information dynamically at the time you open each panel.

To display system information about the current state of the server

- As system administrator, click **Administration Dashboard > Analysis > System Information**.

Panels of the System Information Page

The System Information page contains five panels. When you click a panel to bring it to the front, the page dynamically collects the data for display:

The following table lists the panels, available on the System Information page and their contents.

Panel and Heading	Description
Request/Response	Information that is collected from the user's web request.
Request Information	Typical cgi-bin parameters describing the requested path.
Request Headers	Incoming HTTP headers.
Request Parameters	Incoming HTTP parameters on the URL.
Request Attributes	Attributes (objects) stored on the current request.
Response Information	Miscellaneous information, such as encoding and locale, collected from the request.
Session Misc	User session information.
Session Attributes	Attributes (objects) stored on the user's session.
Locale Information	The current locale of the user.
Presentation Data	Various information used to render requests for this user.
Session Attributes	Portlet Controller Session objects associated with this user.
Request Attributes	Portlet Controller Request objects associated with this user's request.
Application Attributes	Information shared throughout the server (across all users).
System Information	Environment variables, such as Classpath and path.
Server Information	Information about the current front-end server.
Context Information	Servlet object information.

18 Managing My webMethods Server Content

■ Migrating My webMethods Server Content	340
■ Managing Content Storage	340
■ Publishing Portlets as an Administrator	343
■ Rebuilding the Search Index	344
■ Adding Custom JAR Files	345

Migrating My webMethods Server Content

The Content Migration Wizard page enables system administrators to migrate server content from one server instance to another, such as from development to staging to production.

This page can be used to migrate the following types of server content: documents, folders, external links, internal links (using aliases), pages (including layouts), portlets, Dynamic Business Objects (DBOs), permissions, subscriptions, and portlet wiring properties.

Content Migration Considerations

Content migration involves two distinct activities: exporting the content from the source server instance, followed by importing the content on the target server instance. Before performing these actions, consider the following:

- **Migrating portlets and DBOs:** If you are developing or installing any portlets on a development server and want to migrate pages that contain instances of these portlets, you must deploy them on the target server before migrating any pages or published instances of the portlets that were developed or installed on the development server instance.
- **Migrating published content:** Content published to the Content Management system can be migrated from one server instance to another. If you are using your development environment to configure permissions on items published to the content management system, you have the option to migrate the permissions as well.
- **Migrating links:** To properly migrate internal links with references to other objects, such as a link from one page to another, create aliases for these links instead of using the base URL.

For example, if you want to publish a link to an existing page, such as a Sales page that has the following as its initial URL:

```
http://server/meta/default/folder/0000002132
```

Create an alias that points to this URL, but has a more friendly URL such as the following:

```
http://server/Sales
```

- **Migrating permissions and subscriptions:** To properly migrate permissions and subscriptions from a source to target server instance, be sure that both servers are pointing to the same directory services.

Managing Content Storage

System administrators use the **Content Service** page to manage the storage locations available for publishing content to the server. Published content is physically stored in the locations configured in the content service. It typically resides on a separate file server for backup purposes and to provide failover capability and high availability in a My webMethods Server cluster.

My webMethods Server stores the binary content of a file in a content service (either system or network) when you:

- Publish a file to the server
- Update a file
- Check in a new version of a versioned file

The binary content is physically stored in the content service for as long as the file object exists in My webMethods Server. The binary content is marked for deletion when you delete the corresponding file object on My webMethods Server, but the actual file is not deleted until the regularly scheduled purging period. Typically, My webMethods Server purges deleted objects at 2 A.M. server time.

When you set a new default content service, all new content is directed to that service, but content stored in an existing service continues to be accessed from that service.

Content Services

The default system content service stores files in the My webMethods Server database. You cannot modify the system content service configuration.

If users of My webMethods Server store large files (1 MB or larger) or a large number of files, the system content service may not be adequate for your file storage needs. In such cases, you should configure a network content service to provide higher capacity. My webMethods Server supports network file storage by the use of a file system.

On Windows, the file system content service uses a UNC (Uniform Naming Convention) path to connect to a network file storage device. On UNIX, the network file storage must be mounted as a local resource. To support cluster high availability, the network file storage device should provide failover support.

For information on configuring a network content service, see [“Configuring a New Content Service” on page 341](#).

Configuring a New Content Service

This procedure applies only to My webMethods Server instances from on-premise installations. You cannot configure a new content service when My webMethods Server is running in a Docker container.

➤ To configure a new Content Service for the server repository

1. As system administrator: **Administration Dashboard > Content > Content Service > Create New Content Service**.
2. In the **Service Name** field, type a name for your new content service and click **Next**.

The service name is limited from 1 to 255 characters and can contain only alphanumeric ASCII characters with no spaces.

3. Type a physical storage location for your content service.

On Windows, use a valid UNC path. The path is not case-sensitive. For example:

```
\\server\volume\directory\
```

On UNIX, use the path to the location of a mounted network storage, for example:

```
/mounted_location/folder
```

Note:

The sample paths assume that your network administrator has provided the proper security settings to allow all servers in a cluster to have read/write access to the network file system. If My webMethods Server runs as a Windows service, make sure the user account from which you run the service has the required access privileges for the network file system.

4. Click **Submit**.
5. Optionally, to make the new content service the default content service, click  and then click **Set As Default**.

Importing Content from a Content Service

> To import content from an existing content service

1. As system administrator: **Administration Dashboard > Content > Content Service > View Content Services**.
2. Locate the content service to which you want to migrate the contents of an existing content service, click  and then click **Import Content**.
3. For the **Target Folder** property, do one of the following:
 - **Browse**, and then move the target page to the **Selected Items** box and click **Select**.
 - **Use Alias**, and then type the alias of the page to which the user should be redirected to the **Alias Name**. Click **Test** to determine if the alias is valid and the alias target is the correct one. If the alias is correct, click **Select**.
4. Click **Import**.

Setting the Maximum Size for Content

> To set the maximum file size for content published to the server repository

1. As system administrator: **Administration Dashboard > Content > Content Service > Set Publish Constraints.**
2. In the **Max Publish Size (MB)** field, type a maximum publish size (in Megabytes).
3. Click **Apply**.

Specifying Allowed File Extensions for Content

By default, there are no restrictions on the type of file that can be stored in the server repository. However, it is possible to limit the file types that can be stored.

➤ To specify the allowed file extensions for content

1. As system administrator: **Administration Dashboard > Content > Content Service > Set Publish Constraints.**
2. In the **Allowed File Extensions** field, enter a comma-delimited list of file extensions that are allowed.

The default value of * allows the use of any file type. As an example of allowed file extensions, you might see the following list:

zip,doc,xsl,ppt,pdf,gif,jpg,png

3. Click **Apply**.

Publishing Portlets as an Administrator

The Publish page provides system administrators with expanded publishing capabilities that are generally not exposed to most users. The Publish page allows administrators to publish many different types of content such as files, folders, forms, links, and specific portlet instances. You can also publish custom content types such as Dynamic Business Objects or Custom Forms from this page.

➤ To publish content using the Publish page

1. As system administrator: **Administration > Content > Publish.**
2. Select the corresponding option for the content type you wish to publish. The default options are: File, Folder, Form (for DBOs only), Link, and Portlet.
3. For a given content type, select one of the options from the drop down menu.

Note:

If you previously created any custom objects that are based on any given content type, they will now show up as options in the drop down menu for the respective content type. As an example, the RSS Feed option under the Folder content type is a Dynamic Business Object. It was created to extend the Folder object type with custom attributes and business logic for publishing RSS syndicated news feeds to a folder object type.

4. Click **Next**.
5. From the Location heading, click **Browse** to select a parent folder location for the content item you are publishing.

Note:

You can optionally click **Use Alias** if you want to publish the content item to a location that is referenced by an existing alias.

6. Click **Next**.
7. Enter a name for the content item you are publishing.
8. (Optional) Enter a description for the content item you are publishing.
9. Depending on the type of content you are publishing, fill in any Extended Properties for the given content type (such as. RSS Feed URL for an RSS Feed content item).
10. Click **Next**.
11. Click **Finish**.

Rebuilding the Search Index

The Search Administration page allows system administrators to rebuild the search index for the Lucene search engine that is offered with My webMethods Server.

Rebuilding the search index will re-index all content that was previously published to the server and update the default search indexes again. A system administrator might need to do this if the search index somehow becomes corrupted and search stops working.

Tip:

If your server has a lot of content published to the content management system, this operation can take a long time to run. You should run this operation at off-peak hours.

➤ **To rebuild the search index**

1. As system administrator: **Administration > Configuration > Search Administration**.
2. Click **Start Rebuild**.

Adding Custom JAR Files

My webMethods Server runs in an Open Services Gateway initiative (OSGi) framework. You can add custom JAR files to a dedicated directory of the server instance, and run `mws update` to convert those JAR files into OSGi bundles. During the execution of the update command, My webMethods Server generates bundle manifests with default content for the JAR files. If a JAR file in the dedicated *Software AG_directory* /MWS/lib directory already includes an OSGi bundle manifest, My webMethods Server skips the manifest generation for that file.

Additionally, you can customize the default manifest contents by providing instructions in a bind file, for example, when adding a JAR file as a fragment to another bundle.

The following procedure describes how to add custom JAR files and how to attach a JAR file as a fragment bundle. For more information on bundles, see the *OSGi Service Platform Core Specification*.

➤ To add custom JAR files

1. Copy the JAR file to this location:

Software AG_directory /MWS/lib

2. Optional. To include a custom manifest, or add the JAR file as a fragment to another bundle, do the following:
 - a. Create a file with the name *jar_file_name*.bnd where *jar_file_name* is the name of the custom JAR file, without the .jar extension.
 - b. In the bind file, add the custom instructions on how to bind the JAR file. For example:

```
# attach as fragment to the caf.server bundle
Fragment-Host: com.webmethods.caf.server
```

- c. Place the bind file in the same directory where you placed the JAR file:

Software AG_directory /MWS/lib

3. Run the update command for the server instance.

```
Software AG_directory/MWS/bin/mws.[bat | sh] update
```


19 Managing the User Interface

■	Locale Administration	348
■	What Are Server Rules?	348
■	Creating Locale Rules	349
■	Creating Login Page Rules	351
■	Creating Start Page Rules	354
■	Creating Rendering Rules	356
■	Modifying a Rule	358
■	Copying a Rule	360
■	Managing the Evaluation Order for Rules	360
■	Removing a Rule	361
■	Managing Skin Rules	361
■	Managing Shell Rules	364

Locale Administration

My webMethods Server uses several decision points to determine the locale for a user, in this order:

1. The locale preference of the user, if specified in the User Profile. For more information, see [“User Information” on page 153](#).
2. The Locale Rules. For more information, see [“Creating Locale Rules” on page 349](#)
3. The browser locale preference.
4. The system locale of the computer where the server is running, which is the default.

The Locale Administration page allows you to set the default locale rules for My webMethods Server when they are not determined by other rules, as described in [“Creating Locale Rules” on page 349](#).

Note:

If you want to run My webMethods Server as a daemon service in a Linux environment, in the `Software AG_directory/profiles/MWS_instanceName/bin/sagmws_version` script file, set the locale in the LANG variable, for example: `export LANG=es_ES.UTF-8`.

➤ To set the default My webMethods Server locale rule

1. As system administrator: **Administration > User Interface > Locale Administration**.
2. From the **Locale** list, choose the locale to be used as the default rule for My webMethods Server.
3. Click **Apply**.

What Are Server Rules?

My webMethods Server uses rules to control a variety of user activities, from which page they use to log into the server, to the appearance of the pages they see. Using rules, you can define default behaviors for the entire application or you can dynamically control the experience of a given user, group, or role.

The following table describes the types of rules that you can create in My webMethods.

Rule type	Description
Locale rules	Rules that allow you to dictate what locale should be used for a user session if it is not defined in the user profile. For more information about creating a locale rule, see “Creating Locale Rules” on page 349 .

Rule type	Description
Login page rules	Rules that determine what login page should be used. You can, for example, redirect users to different login pages, depending on whether they are inside or outside the firewall. For more information about creating a login page rule, see “Creating Login Page Rules” on page 351 .
Start page rules	Rules that determine what start page should be used. The start page is the page to which the server redirects users after log in. For more information about creating a start page rule, see “Creating Start Page Rules” on page 354 .
Rendering rules	Rules that determine what renderer should be used. <i>Renderers</i> are user interface formatting capabilities that can be assigned to specific server objects by defining rendering rules. You can define rendering rules for virtually any server object type. Rendering rules are useful in providing a consistent look and feel for common object types that can be invoked through explicit rule definitions. For more information about creating a rendering rule, see “Creating Rendering Rules” on page 356 .
Skin rules	Rules that determine what skin should be used. A <i>skin</i> is an installable My webMethods Server component that defines the look and feel of the user interface. Skin rules define what skin should be displayed for a given user, group, or server resource. For example, if a server serves both employees and customers, and there are requirements for a different set of graphics, colors, and fonts for each distinct user population, you can use skin rules to assign the corresponding skin to a given user group. For more information about creating a skin rule, see “Creating Skin Rules” on page 362 .
Shell rules	Rules that determine what shell should be used. A <i>shell</i> is an installable component that generates the My webMethods Server header, footer, and portlet title bars. Shell rules define what shell elements should be displayed for a given user, group, or role. For example, if a server serves both employees and customers, you can use shell rules to assign the corresponding shell to a given user group. For more information about creating a shell rule, see “Creating Shell Rules” on page 364 .

In addition, you can use rules for the creation of roles, which are collections of users, groups, and other roles. You can create a rule-based role that defines members based on the same types of criteria as are used for the rule types. For information, see [“Adding a Rule-Based Role” on page 188](#).

Creating Locale Rules

The Manage Locale Page Rules page allows you to define rules that dictate what locale should be used for a user session if it is not defined in the user profile.

➤ **To create a new locale rule**

1. As system administrator: **Administration > User Interface > Manage Locale Rules > Create New Rule**.
2. In the **Name** field, type a name for the rule.
3. (Optional) In the **Description** field, type a description for the new rule.
4. Unless you want to disable the rule during creation, leave **Is Enabled** selected.
5. To add a condition for individual users, click **Current User(s)** and do the following:
 - a. In the **Keywords** field, type a keyword representing the users you want to search for, and click **Search**.
 - b. Move one or more users to the **Selected** box and click **Apply**.
6. To add a condition based on group or role membership, click **Group / Role Membership** and do the following:
 - a. Under **Search For**, choose the **Groups** or **Roles** option.
 - b. In the **Keywords** field, type a keyword representing the groups or roles you want to search for, and click **Search**.
 - c. Move one or more groups or roles to the **Selected** box and click **Apply**.
7. To add a condition based on user attributes, click **User Attributes**, in the **Pick User Attribute** list, choose a user attribute, and click **Apply**.

For more information about the user attributes in the list, see [“User Information” on page 153](#).
8. To add a condition based on global session attributes, click **Global Session** and do the following:
 - a. In the **Global Session Attribute Name** field, type the name of the global session attribute.
 - b. (Optional) In the **Variable Value** field, type a value for the global session attribute.
 - c. Click **Apply**.
9. To add a condition based on the request header, click **Request** and do the following:
 - a. Choose the expression, and click **Submit**.
 - b. In the **Condition** field, complete the expression.

For example, assume that you want to match any HTTP GET request. The wizard moves `#{request.method}` into the **Condition** field and you type the remainder:

```
#{request.method} == "GET"
```

10. To add a condition that matches the current resource or a parent of the current resource, click **Parent Resource** and then do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
11. To add a condition that matches the current resource type, click **Current Resource Type**, choose the resource type and do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
12. To add a condition for a resource property and a value associated with it, click **Resource Properties**, and do the following:
 - a. In the **Property Name** field, type the property name.
 - b. In the **Property Value** field, type the property value.
 - c. Click **Apply**.

For example, if you want to match files that are PDFs, the property name is `mimeType` and the property value is `pdf`.
13. From the **Result** list, choose the locale to be used for the rule.
14. Click **Create Rule**.

Creating Login Page Rules

The Manage Login Page Rules page allows you to define rules that dictate what login page should be used. Login page rules can be defined to dynamically set the default login page for a given user, group, or role.

➤ **To create a new login page rule**

1. As system administrator: **Administration > User Interface > Manage Login Page Rules > Create New Rule.**
2. In the **Name** field, type a name for the rule.
3. (Optional) In the **Description** field, type a description for the new rule.
4. Unless you want to disable the rule during creation, leave **Is Enabled** selected.
5. To add a condition for individual users, click **Current User(s)** and do the following:
 - a. In the **Keywords** field, type a keyword representing the users you want to search for, and click **Search**.
 - b. Move one or more users to the **Selected** box and click **Apply**.
6. To add a condition based on group or role membership, click **Group / Role Membership** and do the following:
 - a. Under **Search For**, choose the **Groups** or **Roles** option.
 - b. In the **Keywords** field, type a keyword representing the groups or roles you want to search for, and click **Search**.
 - c. Move one or more groups or roles to the **Selected** box and click **Apply**.
7. To add a condition based on user attributes, click **User Attributes**, on the **Pick User Attribute** list, choose a user attribute, and click **Apply**.

For more information about the user attributes in the list, see [“User Information” on page 153](#).
8. To add a condition based on global session attributes, click **Global Session** and do the following:
 - a. In the **Global Session Attribute Name** field, type the name of the global session attribute.
 - b. (Optional) In the **Variable Value** field, type a value for the global session attribute.
 - c. Click **Apply**.
9. To add a condition based on the request header, click **Request** and do the following:
 - a. Choose the expression, and click **Submit**.
 - b. In the **Condition** field, complete the expression.

For example, assume that you want to match any HTTP GET request. The wizard moves `#{request.method}` into the **Condition** field and you type the remainder:

```
#{request.method} == "GET"
```

10. To add a condition that matches the current resource or a parent of the current resource, click **Parent Resource** and then do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
11. To add a condition that matches the current resource type, click **Current Resource Type**, choose the resource type and do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
12. To add a condition for a resource property and a value associated with it, click **Resource Properties**, and do the following:
 - a. In the **Property Name** field, type the property name.
 - b. In the **Property Value** field, type the property value.
 - c. Click **Apply**.

For example, if you want to match files that are PDFs, the property name is `mimeType` and the property value is `pdf`.
13. In the **Result** field, type an alias for the login page, or to browse to the page, click **Pick** and do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
14. Click **Create Rule**.

Creating Start Page Rules

The Manage Start Page Rules page allows system administrators to define rules that dictate what start page should be used. The start page is the page to which the server redirects users after log-in. Start page rules can be defined to dynamically set the default start page for a given user, group, or role.

> To create a new start page rule

1. As system administrator: **Administration > User Interface > Manage Start Page Rules > Create New Rule.**
2. In the **Name** field, type a name for the rule.
3. (Optional) In the **Description** field, type a description for the new rule.
4. Unless you want to disable the rule during creation, leave **Is Enabled** selected.
5. To add a condition for individual users, click **Current User(s)** and do the following:
 - a. In the **Keywords** field, type a keyword representing the users you want to search for, and click **Search**.
 - b. Move one or more users to the **Selected** box and click **Apply**.
6. To add a condition based on group or role membership, click **Group / Role Membership** and do the following:
 - a. Under **Search For**, choose the **Groups** or **Roles** option.
 - b. In the **Keywords** field, type a keyword representing the groups or roles you want to search for, and click **Search**.
 - c. Move one or more groups or roles to the **Selected** box and click **Apply**.
7. To add a condition based on user attributes, click **User Attributes**, in the **Pick User Attribute** list, choose a user attribute, and click **Apply**.

For more information about the user attributes in the list, see [“User Information” on page 153](#).

8. To add a condition based on global session attributes, click **Global Session** and do the following:
 - a. In the **Global Session Attribute Name** field, type the name of the global session attribute.

- a. To find children of a resource, click the name of the resource.
- b. To select a resource, click the option button to the left of the resource.
- c. Click **Apply**.

14. Click **Create Rule**.

Creating Rendering Rules

The Manage Rendering Rules page allows system administrators to configure rendering rules for specific server objects, such as a folder, page, portlet, and so forth. For example, an administrator who wants all folders to display a detailed view of content can create a rendering rule that applies a “details” renderer to all folder objects. For each of the types of conditions you can apply in the following procedure, you can add multiple instances, one at a time.

> To create a new Rendering rule

1. As system administrator: **Administration > User Interface > Manage Rendering Rules > Create New Rule**.
2. In the **Name** field, type a name for the rule.

Example: `folder-thumbnails view` (for image files).
3. (Optional) In the **Description** field, type a description for the new rule.
4. Unless you want to disable the rule during creation, leave **Is Enabled** selected.
5. To add a condition for individual users, click **Current User(s)** and do the following:
 - a. In the **Keywords** field, type a keyword representing the users you want to search for, and click **Search**.
 - b. Move one or more users to the **Selected** box and click **Apply**.
6. To add a condition based on group or role membership, click **Group / Role Membership** and do the following:
 - a. Under **Search For**, choose the **Groups** or **Roles** option.
 - b. In the **Keywords** field, type a keyword representing the groups or roles you want to search for, and click **Search**.
 - c. Move one or more groups or roles to the **Selected** box and click **Apply**.

7. To add a condition based on user attributes, click **User Attributes**, on the **Pick User Attribute** list, choose a user attribute, and click **Apply**.

For more information about the user attributes in the list, see [“User Information” on page 153](#).

8. To add a condition based on global session attributes, click **Global Session** and do the following:
 - a. In the **Global Session Attribute Name** field, type the name of the global session attribute.
 - b. (Optional) In the **Variable Value** field, type a value for the global session attribute.
 - c. Click **Apply**.

9. To add a condition based on the request header, click **Request** and do the following:

- a. Choose the expression, and click **Submit**.
- b. In the **Condition** field, complete the expression.

For example, assume that you want to match any HTTP GET request. The wizard moves `#{request.method}` into the **Condition** field and you type the remainder:

```
#{request.method} == "GET"
```

10. To add a condition that matches the current resource or a parent of the current resource, click **Parent Resource** and then do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
11. To add a condition that matches the current resource type, click **Current Resource Type**, choose the resource type and then do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.

12. To add a condition for a resource property and a value associated with it, click **Resource Properties**, and do the following:
 - a. In the **Property Name** field, type the property name.

- b. In the **Property Value** field, type the property value.
- c. Click **Apply**.

For example, if you want to match files that are PDFs, the property name is `mimeType` and the property value is `pdf`.

13. On the **Result** list, choose the target renderer.

The renderer you select will be applied to all server objects that meet the evaluation criteria you define in the following steps. For example, the thumbnails renderer is useful for displaying thumbnail views for images that are published to the server.

14. Click **Create Rule**.

Modifying a Rule

After a rule exists, you can modify any portion of it that is editable.

> To modify a rule

1. As system administrator: **Administration > User Interface > Manage *rule-type* Rules > View Rules**.

where ***rule-type*** contains the rule you want to modify.

2. For the rule you want to modify, click  and then click **Modify Rule**.
3. Do any of the following:
 - In the **Name** field, type a new name for the rule.
 - (Optional) In the **Description** field, type a new description for the rule.
 - To disable the rule, clear **Is Enabled**.
 - To modify a condition for individual users, click **Current User(s)** and do the following:
 1. In the **Keywords** field, type a keyword representing the users you want to search for, and click **Search**.
 2. Move one or more users to the **Selected** box and click **Apply**.
 - To modify a condition based on group or role membership, click **Group / Role Membership** and do the following:
 1. Under **Search For**, choose the **Groups** or **Roles** option.

2. In the **Keywords** field, type a keyword representing the groups or roles you want to search for, and click **Search**.
 3. Move one or more groups or roles to the **Selected** box and click **Apply**.
- To modify a condition based on user attributes, click **User Attributes**, in the **Pick User Attribute** list, choose a user attribute, and click **Apply**.

For more information about the user attributes in the list, see [“User Information” on page 153](#).

- To modify a condition based on global session attributes, click **Global Session** and do the following:
 1. In the **Global Session Attribute Name** field, type the name for the global session attribute.
 2. (Optional) In the **Variable Value** field, type a new value for the global attribute.
 3. Click **Apply**.
- To modify a condition based on the request header, click **Request** and do the following:
 1. Choose the expression, and click **Submit**.
 2. In the **Condition** field, complete the expression.

For example, assume that you want to match any HTTP GET request. The wizard moves `#{request.method}` into the **Condition** field and you type the remainder: `#{request.method} == "GET"`

- To modify a condition that matches the current resource or a parent of the current resource, click **Parent Resource** and then do the following:
 1. To find children of a resource, click the name of the resource.
 2. To select a resource, click the option button to the left of the resource.
 3. Click **Apply**.
- To modify a condition that matches the current resource type, click **Current Resource Type**, choose the resource type and do the following:
 1. To find children of a resource, click the name of the resource.
 2. To select a resource, click the option button to the left of the resource.
 3. Click **Apply**.
- To modify a condition for a resource property and a value associated with it, click **Resource Properties**, and do the following:
 1. In the **Property Name** field, type the property name.
 2. In the **Property Value** field, type the property value.
 3. Click **Apply**.

- Modify the **Result** as needed field as needed.

4. Click **Update Rule**.

Copying a Rule

If you want to create a rule that is similar to an existing one, you can do so by copying the existing rule.

> To copy a rule

1. As system administrator: **Administration > User Interface > Manage *rule-type* Rules > View Rules**.

where ***rule-type*** contains the rule you want to copy.

2. For the rule you want to copy, click  and then click **Copy Rule**.

3. Type a name for the new rule.

4. (Optional) Type a description for the new rule.

5. Click **Copy The Rule**.

You can then modify the new rule, as described in [“Modifying a Rule” on page 358](#).

Managing the Evaluation Order for Rules

When a user requests a server resource, the server uses rules to determine how to fill the request. For example, perhaps the look and feel of the page is dependent on whether the user is a company employee or a customer. The server evaluates the skin rules to determine which skin to apply, in this order:

1. If there are multiple skin rules, the skin associated with the first rule that matches the user applies.
2. If none of the rules match the user, or if there are no skin rules, the default skin assigned in the **User Preferences** tab of the user’s Profile page applies.
3. If no skin is assigned on the Profile page, the default skin for the server applies.

Changing the Order or Rule Evaluation

If there are multiple skin rules, you can determine the order in which they are evaluated.

> To change the order in rules are evaluated

1. As system administrator: **Administration > User Interface**.
2. Click the name of the set of rules you want to manage.
3. Click **Change Rule Evaluation Order**.
4. To reorder rules, move them up or down as needed.

The first rule in the list is searched first, followed by the second, and so on.

5. Click **Update**.

Removing a Rule

To remove a rule, use the following procedure.

> To remove a rule

1. As system administrator: **Administration > User Interface > Manage *rule-type* Rules > View Rules**.

where *rule-type* contains the rule you want to remove.

2. For the rule you want to remove, click  and then click **Remove Rule**.

Managing Skin Rules

A *skin* is an installable My webMethods Server component that defines the look and feel of the My webMethods Server user interface. A skin modifies the images, fonts, colors, and other subtle stylable aspects of HTML content, but it does not modify the HTML content in any functional way.

A developer creates new custom skins to accomplish many different functions. Some of these include:

- Branding the server with corporate, partner, or departmental logos.
- Aligning the color scheme with corporate, partner, or departmental colors.

Developers create skins with the Skin Administration page and customize them as needed. For more information about how to customize skins, see [“Customizing Skins” on page 433](#).

My webMethods Server offers a variety of ways to configure personalization rules that dictate what skin is displayed for a given user, group, or resource. You can explicitly assign a particular skin to a specific user or set up rules that dynamically assign a skin based on a variety of criteria.

The Manage Skin Rules page allows you to define rules that dictate what skins can be used by users, groups, or roles. This page allows a system administrator to create, modify, or remove rules,

and change the evaluation order of a list of rules that are evaluated for each user every time the user logs in.

Creating Skin Rules

To create a new skin rule, use the following procedure.

> To create a new skin rule

1. As system administrator: **Administration > User Interface > Manage Skin Rules > Create New Rule**.
2. In the **Name** field, type a name for the rule.
3. (Optional) In the **Description** field, type a description for the new rule.
4. Unless you want to disable the rule during creation, leave **Is Enabled** selected.
5. To add a condition for individual users, click **Current User(s)** and do the following:
 - a. In the **Keywords** field, type a keyword representing the users you want to search for, and click **Search**.
 - b. Move one or more users to the **Selected** box and click **Apply**.
6. To add a condition based on group or role membership, click **Group / Role Membership** and do the following:
 - a. Under **Search For**, choose the **Groups** or **Roles** option.
 - b. In the **Keywords** field, type a keyword representing the groups or roles you want to search for, and click **Search**.
 - c. Move one or more groups or roles to the **Selected** box and click **Apply**.
7. To add a condition based on user attributes, click **User Attributes**, in the **Pick User Attribute** list, choose a user attribute, and click **Apply**.

For more information about the user attributes in the list, see [“User Information” on page 153](#).

8. To add a condition based on global session attributes, click **Global Session** and do the following:
 - a. In the **Global Session Attribute Name** field, type the name of the global session attribute.
 - b. (Optional) In the **Variable Value** field, type a value for the global session attribute.

- c. Click **Apply**.
9. To add a condition based on the request header, click **Request** and do the following:
 - a. Choose the expression, and click **Submit**.
 - b. In the **Condition** field, complete the expression.

For example, assume that you want to match any HTTP GET request. The wizard moves `#{request.method}` into the **Condition** field and you type the remainder:

```
#{request.method} == "GET"
```

10. To add a condition that matches the current resource or a parent of the current resource, click **Parent Resource** and then do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
11. To add a condition that matches the current resource type, click **Current Resource Type**, choose the resource type and then do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
12. To add a condition for a resource property and a value associated with it, click **Resource Properties**, and do the following:
 - a. In the **Property Name** field, type the property name.
 - b. In the **Property Value** field, type the property value.
 - c. Click **Apply**.

For example, if you want to match files that are PDFs, the property name is `mimeType` and the property value is `pdf`.
13. On the **Result** list, choose the target skin.
14. Click **Create Rule**.

Managing Shell Rules

A *shell* is an installable component of My webMethods Server. A shell is segment is a special kind of page that generates the My webMethods Server header, footer, and portlet title bars.

Where regular portlets produce the primary content of a page, a shell provides the structure that frames that primary content. Common web page idioms such as banners, standard links, and search boxes appear in a shell.

A developer creates new custom shells to accomplish many different functions, such as:

- Adding a row of links to other corporate web sites below the page banner
- Changing the default search box to one that searches the corporate catalogue
- Adding a left-hand navigation bar to every page.

You can set up many different criteria to determine which shell is used for a particular user request. My webMethods Server offers a variety of ways to configure personalization rules that dictate what shell is displayed for a given user, group, or resource.

Note:

Unlike skins, you cannot explicitly assign a particular shell to a specific user. You need to use rules that dynamically assign a shell based on a variety of criteria.

The Manage Shell Rules page allows you to define rules that dictate what shells can be used by users, groups, or roles. This page allows a system administrator to create, modify, or remove rules, and change the evaluation order of a list of rules that are evaluated for each user every time the user logs in.

Creating Shell Rules

To create a new shell rule, use the following procedure.

> **To create a new shell rule**

1. As system administrator: **Administration > User Interface > Manage Shell Rules > Create New Rule.**
2. In the **Name** field, type a name for the rule.
Example: `folder-thumbnails view` (for image files).
3. (Optional) In the **Description** field, type a description for the new rule.
4. Unless you want to disable the rule during creation, leave **Is Enabled** selected.
5. To add a condition for individual users, click **Current User(s)** and do the following:

- a. In the **Keywords** field, type a keyword representing the users you want to search for, and click **Search**.
 - b. Move one or more users to the **Selected** box and click **Apply**.
6. To add a condition based on group or role membership, click **Group / Role Membership** and do the following:
 - a. Under **Search For**, choose the **Groups** or **Roles** option.
 - b. In the **Keywords** field, type a keyword representing the groups or roles you want to search for, and click **Search**.
 - c. Move one or more groups or roles to the **Selected** box and click **Apply**.
 7. To add a condition based on user attributes, click **User Attributes**, in the **Pick User Attribute** list, choose a user attribute, and click **Apply**.

For more information about the user attributes in the list, see [“User Information” on page 153](#).

8. To add a condition based on global session attributes, click **Global Session** and do the following:
 - a. In the **Global Session Attribute Name** field, type the name of the global session attribute.
 - b. (Optional) In the **Variable Value** field, type a value for the global session attribute.
 - c. Click **Apply**.
9. To add a condition based on the request header, click **Request** and do the following:
 - a. Choose the expression, and click **Submit**.
 - b. In the **Condition** field, complete the expression.

For example, assume that you want to match any HTTP GET request. The wizard moves `#{request.method}` into the **Condition** field and you type the remainder:

```
#{request.method} == "GET"
```

10. To add a condition that matches the current resource or a parent of the current resource, click **Parent Resource** and then do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.

- c. Click **Apply**.
11. To add a condition that matches the current resource type, click **Current Resource Type**, choose the resource type and then do the following:
 - a. To find children of a resource, click the name of the resource.
 - b. To select a resource, click the option button to the left of the resource.
 - c. Click **Apply**.
 12. To add a condition for a resource property and a value associated with it, click **Resource Properties**, and do the following:
 - a. In the **Property Name** field, type the property name.
 - b. In the **Property Value** field, type the property value.
 - c. Click **Apply**.

For example, if you want to match files that are PDFs, the property name is `mimeType` and the property value is `pdf`.

13. In the **Result** list, choose the target shell.

The renderer you select will be applied to all server objects that meet the evaluation criteria you define in the following steps. For example, the thumbnails renderer is useful for displaying thumbnail views for images that are published to the server.

14. Click **Create Rule**.

Setting Shells for Requests

To set a specific shell for a request, a developer creates a link to a server resource and adds a shell parameter to the link. The shell parameter value should be an alias to the target shell.

For example, the URL for a link to the public folder with the extranet shell is `/folder.public?shell=shell.extranet`. When users follow the link, they view the public folder framed with the specified extranet shell. When users click another link from the public folder page, they return to whatever shell they were using before, provided the link does not also have a shell parameter.

Setting Shells for Sessions

To set a specific shell for a session, sometimes referred to as a *sticky* shell because the setting is retained for the duration of the session, a developer creates a link to the `forceShell` command. This

command takes a returnUrl parameter, which redirects a user to the specified URL once the shell has been set.

For example, the URL for a link to the public folder with a sticky extranet shell is `/?command=forceShell&shellURI=shell.extranet&returnUrl=folder.public`. When users follow the link, they view the public folder now framed with the extranet shell. When users click another link from the public folder page, provided the link does not have a shell parameter, they see that page still framed with the extranet shell.

20 Working with the Common Directory Services API

■ Managing User Information with the Common Directory Service API	370
■ About the Common Directory Services API	371
■ Prerequisites	371
■ About CDS Version Interoperability	372
■ CDS Code Examples	373

Managing User Information with the Common Directory Service API

My webMethods Server provides several directory service options for managing users and groups:

- My webMethods system directory. This is an internal My webMethods Server user directory, available by default in all installations of My webMethods Server. You can access information in this directory server using the My webMethods user interface and the Common Directory Services (CDS) API. Both read and write access are available.
- LDAP (Lightweight Directory Access Protocol). My webMethods enables you to define one or more external LDAP user directories. For a list of supported directory server products, see *System Requirements for Software AG Products*. You can access information in this directory server using the My webMethods user interface and the CDS API. Only read access is available.
- Database. My webMethods Server also enables you to authenticate users against a database directory, which is a set of RDBMS tables and an SQL configuration to access these tables. You can implement a custom authentication module to extend authentication against a database directory. You can access information in this directory server using the My webMethods user interface and the CDS API. Only read access is available.

My webMethods Server and applications and services running within it can access the user information contained in these directories, and you can configure external applications and services that have access to My webMethods Server to use this data.

In addition to working with users and groups in a directory service, you can access and maintain role information, which is maintained separately in the My webMethods Server database.

For example, you can:

- Configure other suite applications, such as webMethods Integration Server, to authenticate users from any of the above user directory options instead of from a user directory unique to Integration Server.
- Configure a process step in a business process to call a Java service to obtain user attributes from the directory service or role membership from the My webMethods Server database, and pass that data into the process pipeline.
- Configure a Java service to assign a user to a role programmatically.

The CDS API offers support for the following:

- Search and discovery of users, groups, and roles.
- Support for LDAP search controls for large directories.
- Create and update users and groups in the system directory. All other external directories are read-only.
- Delete users and groups from the system directory.
- Create, update, and delete roles in My webMethods Server.

- Read custom attributes from LDAP and database directories.
- Read and write custom profile attributes for users, groups, and roles (that is, attributes which are not managed by external directories).

For information about the Common Directory Services API, see the `com.webmethods.sc.directory` and `com.webmethods.sc.mws` packages in the *webMethods CAF and My webMethods Server Java API Reference*.

About the Common Directory Services API

When connected to the same My webMethods Server database and Universal Messaging server, CDS behaves very much like a My webMethods Server cluster instance and participates in all distributed caching across the cluster. When a system directory user is updated, the change will be detected by all CDS instances within a few minutes.

Directory services are defined and managed in My webMethods Server, by logging in to My webMethods as either SysAdmin or as Administrator. For more information about working with directory services, see [“Managing Directory Services” on page 110](#) and [“Managing External Directory Services” on page 112](#).

In addition to working with the CDS API, you can use the My webMethods interface to perform all directory management activities:

- User, group, and role management. For more information, see [“About Managing Users and Groups” on page 150](#).
- Authentication management. For more information, see [“Managing Security” on page 251](#).
- Implementation of attribute providers. For more information, see [“Attribute Providers” on page 241](#).

Prerequisites

Before you can work with the CDS API, you must take the following actions:

- Create the database tables for My webMethods Server using the webMethods Database Component Configurator. This is usually carried out immediately after installation.
- Create and initialize the My webMethods Server instance.
- Initialize CDS with a JDBC connection to the My webMethods Server database schema. For more information, see [“Initializing Common Directory Services” on page 372](#).
- Configure a Universal Messaging server to use as a JMS provider from the **Cluster Administration** page in My webMethods after installing My webMethods Server, or when creating a new instance using the My webMethods Server command line utility or Software AG Command Central.

Class Path Considerations

To use CDS in a stand-alone application, in addition to providing the correct JDBC connection information, all CDS .jar files must be present in the class path of the external application running CDS. To ensure that these .jar files are available, include all .jar files from *Software AG_directory* /common/lib and /common/lib/ext (assuming you have a standard installation of My webMethods Server and Integration Server).

Initializing Common Directory Services

When the CDS API is accessed from within My webMethods Server (from a CAF application for example), or from Integration Server, CDS is already initialized and no further action is needed. If you want to use the CDS API from an external application or service that has access to My webMethods Server, the CDS API must be explicitly initialized from the external application or service.

You initialize Common Directory Services by invoking the `com.webmethods.sc.mws.MWSLibrary.init()` static method. The input parameters are expected as Java system properties and must describe a JDBC connection URL to a My webMethods Server database schema. Instead of using remote call backs to My webMethods Server, the CDS API connects to this schema and reads all configuration and principal information.

Here is sample code showing how to initialize CDS using the `MWSLibrary` class:

```
System.setProperty(MWSLibrary.SYSTEM_PROP_DB_DRIVER,
"com.wm.dd.jdbc.sqlserver.SQLServerDriver"); // JDBC Driver Class

System.setProperty(MWSLibrary.SYSTEM_PROP_DB_URL,
"jdbc:wm:sqlserver://localhost:1433;DatabaseName=webm82_dev");
// JDBC Connection URL

System.setProperty(MWSLibrary.SYSTEM_PROP_DB_USER, "webm82_dev");
// DB username

System.setProperty(MWSLibrary.SYSTEM_PROP_DB_PASSWORD, "password");
// DB
password

MWSLibrary.init();
```

About CDS Version Interoperability

My webMethods Server can communicate and exchange user information with a newer version of the Common Directory Services component when the hosting Integration Server instance is from a later release. For example, a My webMethods Server instance with version 10.3 can communicate with the CDS component from an Integration Server installation with version 10.5. The version compatibility mode is not enabled by default. For information about how to enable version interoperability for My webMethods Server and CDS, see [“Enabling Version Interoperability in My webMethods Server” on page 373](#).

Enabling Version Interoperability in My webMethods Server

1. Retrieve the `cluster.xml` file for My webMethods Server using the `getConfig` command.
2. Open the `cluster.xml` file in a text editor, and set the `useInteropMode` attribute in the `<Cluster>` element to `true`, as follows:

```
<Cluster clusterId="ID" frontEndSecurePort="-1"
frontEndUrl="http://localhost:8585" useInteropMode="true">
```

3. Save the file and execute the `putConfig` command to store it to the database.
4. Remove the local copy of the `cluster.xml` file from the node.

For more information about working with the `cluster.xml` file, see [“Working with the cluster.xml File” on page 222](#).

5. Configure Central User Management in Integration Server as described in *webMethods Integration Server Administrator's Guide*.

CDS Code Examples

List All Roles

```
IDirectorySession session =
DirectorySystemFactory.getDirectorySystem().createSession();

List roles = session.listRoles();

for (IDirectoryRole role: roles) {

    String roleID = role.getID();

    String roleName = role.getName();

    String roleDN = role.getDN();

}
```

Lookup a User by Name and Fetch all Attribute

```
IDirectorySession session =
DirectorySystemFactory.getDirectorySystem().createSession();

IDirectoryUser user = (IDirectoryUser) session.lookupPrincipalByName
("user1", IDirectoryPrincipal.TYPE_USER);

Map attributes = user.getAllAttributes();
```

Authenticate User

```
IDirectorySession session =  
DirectorySystemFactory.getDirectorySystem().createSession();
```

```
IDirectoryUser user = session.authenticateUser("username", "password");
```

Create Static Role and Add User as a Member

```
IDirectorySession session =  
DirectorySystemFactory.getDirectorySystem().createSession();
```

```
IDirectoryUser user = (IDirectoryUser) session.lookupPrincipalByName  
("user1", IDirectoryPrincipal.TYPE_USER);
```

```
IDirectoryRole role = session.createRole(IDirectoryRole.STATIC_ROLE_TYPE,  
"roleName", Collections.EMPTY_MAP);
```

```
session.addPrincipalToRole(user.getID(), role.getID());
```

21 Sending Mobile Notifications from My webMethods Server

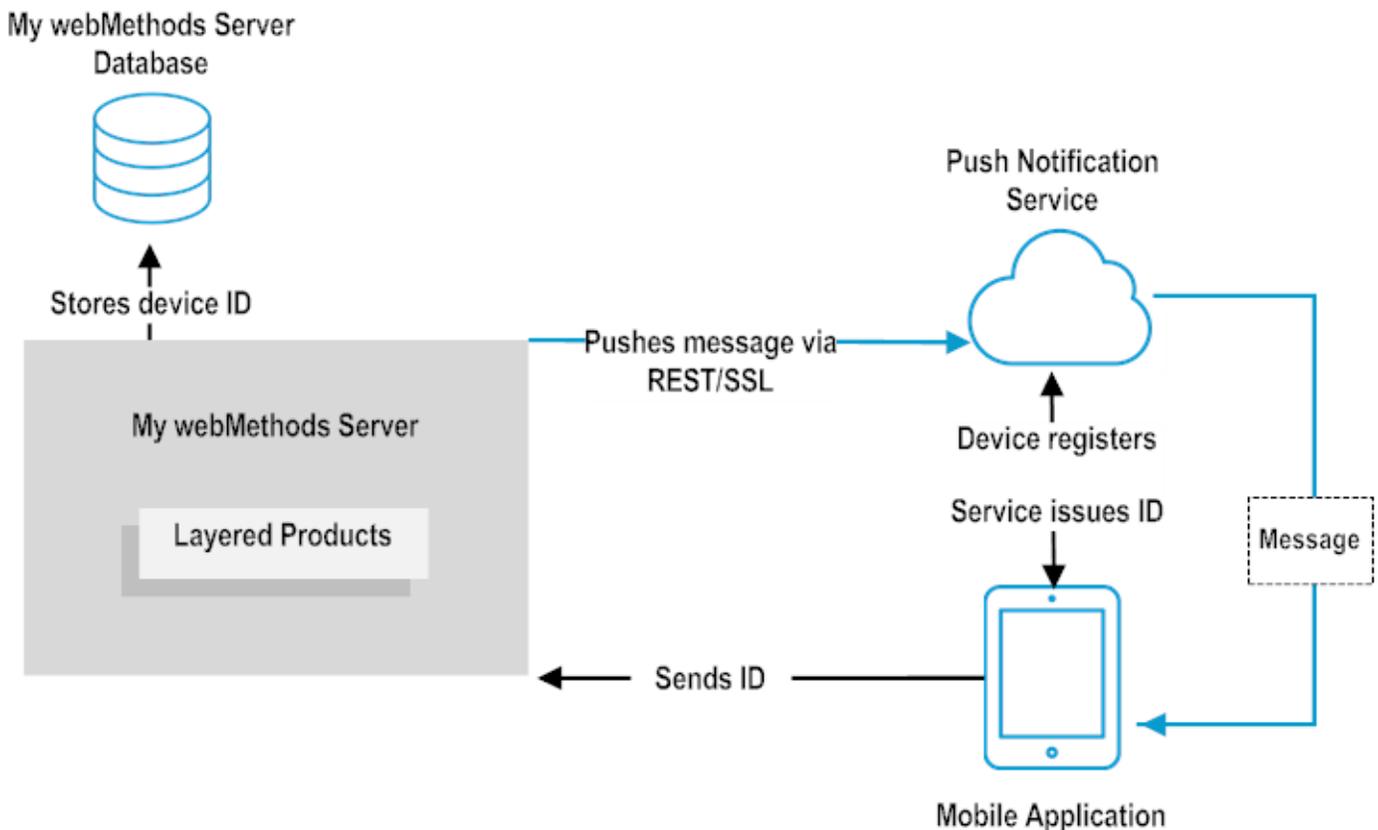
- Sending Push Notifications to Mobile Devices 376
- Configuring Push Notifications in My webMethods Server 377

Sending Push Notifications to Mobile Devices

You can configure My webMethods Server to send push notifications for events to users on mobile devices.

My webMethods Server uses different push notification service providers for sending messages to client applications, depending on the mobile operation system. The default notification provider for Android applications is Google Cloud Messaging, and the default notification provider for Apple is iStore. Before you can send push notification messages from My webMethods Server, you must register your mobile application with the respective service provider.

When a device running the mobile applications registers with the respective push notification provider, the provider issues a registration ID for the device. The device sends the registration ID to My webMethods Server, together with details about the events to which to subscribe. My webMethods Server stores the device ID and event subscription details in the server database. When a relevant event occurs, My webMethods Server pushes a message to the notification service provider, with the device registration ID. The push notification service provider sends the message to the registered device. If a mobile application user uses multiple devices to connect, My webMethods Server sends the notification to each registered device.



My webMethods Server uses REST to push notifications to Google Cloud Messaging, and SSL to push notifications to iStore.

My webMethods Server provides an out-of-the-box implementation to send push notifications for Task Engine events to Mobile Business Console users on mobile devices. With this implementation, My webMethods Server sends a notification each time a user, or a role or group that the user belongs to, is assigned or delegated a task. To send push notifications for a custom mobile application, you must do the following:

- Implement custom event handlers for the event types for which you want to send notifications. For more information, see *webMethods CAF and OpenUI Development Help*.
- Create a custom service to push notifications through the `publishNotification` API in My webMethods Server. For more information, see *webMethods CAF and My webMethods Server Java API Reference*.
- Register the application with the push notification service. For more information about how to register with the push notification service provider, see the provider documentation.
- Configure the mobile push notification options in My webMethods Server, as described in [“Configuring Push Notifications in My webMethods Server” on page 377](#).

Configuring Push Notifications in My webMethods Server

You configure push notifications in My webMethods Server after you register your mobile application with the push notification service for the respective operating system.

1. As sysadmin, navigate to **Folders > Administrative Folders > Administration Dashboard > Configuration > Mobile Push Notification Administration**.
2. Click **Add New App**.
3. On the **Mobile App Config** page, specify the name of your mobile application and the application details.
 - For Android applications:

The following table lists the properties, required for Android applications:

Field	Description
End Point URL	The endpoint of the send API from your registration with Google Cloud Messaging.
OAuth	The OAuth authorization token for your application, as registered with the Google Cloud Messaging.

- For iOS applications:

The following table lists the properties, required for iOS applications:

Field	Description
Host Name	The URL of the Apple Push Notification service from your registration with iStore.
Port Number	The number of the port to use for SSL communication with the iStore push notification service. The default port number is 2195.
Certificate Path	The full path to the location of the SSL certificate from your registration with iStore.
Password	The password to use to connect to iStore.

4. Click **Submit**.

IV Server Page Development

22	Managing Pages in My webMethods Server	381
23	Managing Workspaces in My webMethods Server	411
24	Customizing Skins	433
25	Working with Shells in My webMethods Server	459

22 Managing Pages in My webMethods Server

- About Page Development 382
- About Custom Folders and Pages 382
- About Customizing the My webMethods Navigation 399
- About Customizing the My webMethods Look-And-Feel 404
- Building a Simple Front-End Page to My webMethods 408
- Creating Links for Single Sign-On 409

About Page Development

My webMethods Server enables you to easily create custom pages and have My webMethods Server serve the custom pages as webpages. For more information, see [“About Custom Folders and Pages” on page 382](#).

My webMethods Server also provides a flexible architecture that enables you to customize the My webMethods user interface, as described in the following table.

Activity	Additional Information
Update the My webMethods navigation panel by: <ul style="list-style-type: none"> ■ Adding selections to the navigation panel ■ Removing selections from the navigation panel ■ Hiding standard tabs and sections in the navigation panel ■ Completely replacing the navigation panel 	“About Customizing the My webMethods Navigation” on page 399
Update the My webMethods look-and-feel by: <ul style="list-style-type: none"> ■ Replacing the logo image ■ Changing the color scheme ■ Applying a custom look-and-feel 	“About Customizing the My webMethods Look-And-Feel” on page 404
Build an alternate way to access My webMethods application pages; that is, building a simple front-end page that provides links to only a few My webMethods pages that users require.	“Building a Simple Front-End Page to My webMethods” on page 408

Additionally, you can use the techniques listed above to build complete custom applications that run in My webMethods Server.

About Custom Folders and Pages

You can create custom folders and pages that My webMethods Server serves as webpages. To My webMethods Server both are functionally equivalent; that is, My webMethods Server displays the contents of both as portlets on a webpage. You decide whether to use a folder or page based on how you intend to use it. Use a folder if you want a container that holds other folders and pages. Use a page if you want to display information.

Create custom pages when you need a page that you would consider permanent, that is to be used for a long period of time. If you need a page that is not intended to be permanent and that you can use as a work area, it is more appropriate to use a workspace. For information, see [“Managing Workspaces in My webMethods Server” on page 411](#).

To add content to a page, you drag and drop portlets on to the page. You can position the portlets where you want, size them, and set their properties. You can make the page dynamic by using wiring:

- You can wire the property of one portlet to the property of another. By doing so, when the property in the source portlet is set, the change is automatically reflected in the destination portlet.
- You can wire information about the user accessing the custom page to portlet properties. For example, if you have a portlet that displays the weather, you can wire the postal code attribute from a user profile to the postal code property of the weather portlet. As a result, when the user displays the page, My webMethods Server automatically uses the postal code from the user's profile to set the postal code property of the weather portlet and the user's local weather displays.

Creating Custom Pages

Create portal pages by logging into My webMethods using the SysAdmin user account or another user account that is a member of the Admin role. To build a page, you must switch to page editing mode. In page editing mode, the system administrator user interface changes to display a **Tools** tab on the left and the custom page on the right. The **Tools** tab lists portlets that you can add to the custom page.

➤ To create a custom page

1. As system administrator, navigate to the folder where the new page is to reside. If the folder you want to use does not exist, create it.
 - If you want to add the page to the My webMethods Applications navigation, navigate to a location within:

Folders > My webMethods Applications > Fabric Tasks
 - If you want to add the page to a different taxonomy, it is recommended that you use a location within either of the following:
 - Folders (root folder)
 - **Folders > Public Folders**
2. In the folder title bar, click , and then **New > Page**.
3. In the New Page window, do the following:
 - a. In the **Name** field, type the name of the new page.
 - b. In the **Description** field, optionally type a description of the new page.
 - c. Click **Create**.

4. Open the new page you just created by clicking the link for the page.
5. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
6. Set the page properties. For instructions, see [“Page Properties” on page 385](#) and [“Setting Page Properties” on page 386](#).
7. Define the page layout. For instructions, see [“Controlling the Page Layout” on page 387](#).
8. Build the page by performing the tasks listed in the following table:

Task	Additional Information
Add portlets by dragging portlets from the Tools tab on to the page canvas.	“Adding Portlets to a Page” on page 391
Position the portlets on the page.	“Positioning Portlets on a Page” on page 392
Set the portlet properties.	“Modifying Portlet Properties” on page 395
Optionally, set up portlet aliases.	“Managing Portlet Aliases” on page 396
Optionally, wire the portlets.	“Wiring the Property of One Portlet to the Property of Another” on page 397 and “Wiring a Principal Attribute to a Portlet Property” on page 398

9. In the page title bar, click **Save**.

To make your new page available, you can:

- Provide the page URL so that users can access it directly. Make the URL simpler by assigning the page an alias. For example, if you assign the page the alias “MyCustomPage”, users can enter the following URL where *host_name* is the My webMethods Server host name and *port_number* is its port number.

`http://www.host_name:port_number/MyCustomPage`

For more information about how to assign aliases, see [“Setting Page Properties” on page 386](#).

- Add the page to the My webMethods navigation so that users can select it from there. For more information, see [“Adding Selections to the My webMethods Navigation” on page 400](#).

Editing an Existing Page

After you initially create a page, you can open the page in page editing mode at any time to make additional changes to it.

> To edit an existing page

1. As system administrator, navigate to and open the page you want to edit.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. In page editing mode, do one of the following:

The following table lists editing options for an existing page, and where to find more information about each option:

Task	Additional Information
Update the page properties, for example, to change the name of the page.	“Page Properties” on page 385 and “Setting Page Properties” on page 386
Change the page layout.	“Controlling the Page Layout” on page 387
Add more portlets to the page.	“Adding Portlets to a Page” on page 391
Remove portlets from the page	“Removing Portlets from a Page” on page 391
Reposition the portlets on the page	“Positioning Portlets on a Page” on page 392
Update portlet properties.	“Modifying Portlet Properties” on page 395
Set up portlet aliases.	“Managing Portlet Aliases” on page 396
Wire the portlets.	“Wiring the Property of One Portlet to the Property of Another” on page 397 and “Wiring a Principal Attribute to a Portlet Property” on page 398

4. In the page title bar, click **Save**.

Page Properties

The **General** tab for a page displays the properties, listed in the following table:

Property	Description
Name	The name of the page.
Description	An optional description of the page.
Keywords	Optional keywords that you assign to a page for your own use. Out-of-the-box, My webMethods Server does not provide functionality that uses the keywords. However, you can write custom code that takes advantage of the keywords. For example, you might create custom search code that allows you to search based on the keywords.
Owner	The owner of the page. Click the link to view the owner’s profile.

Property	Description
Created On	The date and time when the page was created.
Modified On	The date and time the page was last updated.
Aliases	Aliases assigned to the page.

For instructions for how to set the properties, see [“Setting Page Properties” on page 386](#). For information about the **Layout** tab, see [“Controlling the Page Layout” on page 387](#).

Setting Page Properties

Use the following procedure to set properties for a page. For a description of the properties, see [“Page Properties” on page 385](#).

> To set page properties

1. As system administrator, navigate to and open the page for which you want to set properties.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. Click **Properties**.
4. On the **General** tab, specify:

The following table lists the page properties, available on the General tab and their functions:

Task	Property	Action
Change the name of the page	Name	Type a new name in the Name field.
Change or add a description of the page	Description	Type a description in the Description .
Assign keywords to the page	Keywords	In the Keywords field, type one or more keywords separated by commas.
Assign aliases to the page	Aliases	<p>To add an alias:</p> <ol style="list-style-type: none"> a. Click Add. b. In the text box, type the alias name you want to add. c. Click OK. <p>To update an alias:</p> <ol style="list-style-type: none"> a. Select the alias you want to change.

Task	Property	Action
		b. Click Edit . c. In the text box, type the updated alias name. d. Click OK . To remove an alias: a. Select the alias you want to remove. b. Click Remove .
5.		Click OK .
6.		In the page title bar, click Save .

Controlling the Page Layout

To define how the portlets within a page can be positioned, define the page layout. There are two types of layouts:

- **Column layout.** In this layout, all the portlets in a page are aligned in columns. Portlets cannot overlap within a column. You can define a page to have one, two, three, or four columns. The default for a new page is a two column layout.

Each column has a single row. You can add additional rows to a column by dragging the **Row** tool into the column. For more information about using rows, see [“Adding Rows When Using a Column Layout” on page 390](#) and [“Removing Rows When Using a Column Layout” on page 390](#).

- **Free Form layout.** In this layout, you can place portlets on the page in any location. Portlets can overlap and even completely cover one another.

For instructions for adding portlets to a page, see [“Adding Portlets to a Page” on page 391](#). For instructions for how to position portlets within a page, see [“Positioning Portlets on a Page” on page 392](#).

➤ To define the layout of a page

1. As system administrator, navigate to and open the page for which you want to define the layout.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. Click **Properties**.

4. Select the **Layout** tab.
5. Use the **View As** list to indicate whether you want to view column, row, and portlet borders and hidden portlet title bars.
 - Select **End User** if you do *not* want to view borders or hidden title bars.
 - Select **Expert User** if you want to view borders and hidden title bars.

Borders are helpful when you are positioning portlets. Also, if you set a portlet's properties so that the title bar is hidden, you must view the page as an expert user to re-display the title bar so that you can take action on that portlet, for example, to move it or access its properties.

Note:

The setting to view as an expert is temporary. If you leave the edited page, when you return, the view will be as an end user again.

Note:

The **View As Expert** check box at the top of the page serves the same purpose as the **View As** property. If you select **Expert User** in the **View As** property, My webMethods Server automatically selects the **View As Expert** check box when you save the properties. Similarly, if you select **End User** in the **View As** property, My webMethods Server automatically clears the **View As Expert** check box when you save the properties.

6. Set the **Editable Canvas** property based on whether you want users to be able to reposition and/or resize portlets on the page.
 - Select the **Editable Canvas** check box if you want users to be able to reposition and resize portlets on the page.
 - Clear the **Editable Canvas** check box if you do not want users to be able to reposition and resize portlets on the page.

Clearing the **Editable Canvas** check box prevents end users from inadvertently changing the layout while using the page. If you select a free form layout for the page, a user can inadvertently change the page simply by clicking a portlet.

Note:

When the **Editable Canvas** check box is selected, users can change the layout. This is true even when users are denied edit permissions, although they are prevented from saving those edits. However, when the **Editable Canvas** check box is cleared, all users are prevented from changing the layout.

7. From the **Columns** list, select layout that you want to use for the page.
8. If you select **One Column**, **Two Columns**, **Three Columns**, or **Four Columns**, you can set additional properties for each column in the layout.

- a. In the **Attributes** field, optionally specify attributes for the column for your own use. Out-of-the-box, My webMethods Server does not provide functionality that uses the attributes. However, you can write custom code that takes advantage of the attributes.
- b. In the **Width** field type the percentage of the page to use for the column. By default the percentages are set for evenly spaced columns.
- c. Select the **Word Wrap** check box if you want the server to attempt to wrap long text lines within portlets to fit the column size. Clear the check box if you do not want long lines to wrap.

Allowing long lines to wrap helps to better fit portlets within columns.

- d. From the **Horizontal Alignment** list select how the portlets should be aligned horizontally in the column. By default, the server left aligns the portlets in each column.
- e. From the **Vertical Alignment** list select how the portlets should be aligned vertically in the column. By default, the server aligns the portlets at the top of each column.
- f. To apply a CSS class to the column, in the **CSS Class** field, type the name of the class, omitting the leading period. For example, type nav for the ".nav" class.

Specify a CSS class defined by a CSS style sheet included in the page, either the CSS style sheet that is:

- Used by the current skin
 - Included with a custom portlet in the page content or the current shell
- g. To apply a style to the column, in the **CSS Style** field, type any style that is valid for use in a CSS file. For example, if you type border: 1pt dashed red, a dashed red line appears as a column border.
 - h. To apply a background image to the column, in the **Skin Background Image** field, type the name of the image file from the current skin.

You specify the name of the skin property. For example, to use the main logo image from the skin, type images/logo.gif. Determine the skin property name for an image by accessing the skin editor (via the **Administration Dashboard > User Interface > Skin Administration** portlet) and locating the image in the Images page. The skin property is "images/" followed by the property name, such as "images/logo.gif".

Note:

To apply an image that is not in the current skin, specify the standard CSS background-image property in the **CSS Style** field. You can also specify properties in the **CSS Style** field to control how the background is displayed, such as whether it repeats, is centered, scrolls with the page, etc.

9. Click **OK**.

10. In the page title bar, click **Save**.

Adding Rows When Using a Column Layout

By default, when you set the layout of a page to use a column layout, the page has a single row in each column. The portlets you add to a column are aligned vertically based on the vertical alignment you specify for the column. However, you might want to add additional rows to one or more columns.

> To add a row to a column on a page

1. As system administrator, navigate to and open the page to which you want to add a row.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. Select the **View As Expert** check box so that you can see the borders and title bars of the rows that you add to the page.
4. In the **Tools** tab, expand the **Layout** item to reveal the **Row** tool.
5. Drag the **Row** tool into the column where you want it.

The system displays a red box beneath the cursor position to indicate where the row would be positioned if you released the mouse button.

6. Click **Save**.

Removing Rows When Using a Column Layout

If you added rows to a column in the layout and no longer want the row, you can remove it.

> To remove rows from a column

1. As system administrator, navigate to and open the page from which you want remove a row.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. Select the **View As Expert** check box so that you can see the borders and title bars of the rows on the page.
4. If the row contains any portlets that you want to preserve, drag them out of the row into a to a different location on the page.
5. In the title bar of the row, click .

6. Click **Save**.

Adding Portlets to a Page

You can add as many portlets as you want to a page. You can also add the same portlet multiple times. Where you can position a portlet on a page depends on the page layout. For more information, see [“Controlling the Page Layout” on page 387](#) and [“Positioning Portlets on a Page” on page 392](#).

➤ To add portlets to a page

1. As system administrator, navigate to and open the page to which you want to add portlets.
2. In the page title bar, click  **Edit Page** to switch to page editing mode.

In page editing mode, the **Tools** tab on the left lists the portlets you can add to a page. Many of the portlets listed in the **Tools** tab are described in the *My webMethods Server Portlet Reference*.

3. If you want to view the column, row, and portlet borders to help with the placement of your portlets, select the **View As Expert** check box. Clear the check box when you no longer want to view the borders.
4. In the **Tools** tab select the portlet you want to add and drag it on to the page.

The system displays a red box beneath the cursor position to indicate where the portlet would be positioned if you released the mouse button.

5. Click **Save**.

Removing Portlets from a Page

To remove a portlet from a page, use the following procedure.

➤ To remove a portlet from a page

1. As system administrator, navigate to and open the page from which you want to remove portlets.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. If the title bar of the portlet you want to remove is hidden, select the **View As Expert** check box so that you can see the title bar.
4. In the title page of the portlet that you want to delete, click , and then **Delete**.

5. Click **Save**.

Positioning Portlets on a Page

Where you can position portlets on a page depends on the page layout.

- When using a column layout (e.g., Three Columns), the layout forces the portlets to be aligned within the columns. You can move portals up and down within a column or from one column to another.
- When using a free form layout, you can move portlets anywhere within the page. Portlets do not have to be aligned and can overlap one another.

For instructions for how to define the page layout, see [“Controlling the Page Layout” on page 387](#).

➤ To position a portlet on a page

1. As system administrator, navigate to and open the page you want to update.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. If the title bar of the portlet you want to reposition is hidden, select the **View As Expert** check box so that you can see the title bar.
4. Move your cursor over the title bar of the portlet you want to reposition until the system displays the move cursor.
5. Click and drag the portlet to the new location.

The system displays a red box beneath the cursor position to indicate where the portlet would be positioned if you released the mouse button.

6. Click **Save**.

Portlet Properties

General Tab

My webMethods Server displays portlet properties in regular view mode and in page editing mode. The following table list the portlet properties, available on the **General** tab.

Section	Property	Description
General	Name	Name of the portlet that appears in the portlet title bar.
	Description	Optional description of the portlet.

Section	Property	Description
	Keywords	Optional keywords that you assign to a portlet for your own use. Out-of-the-box, My webMethods Server does not provide functionality that uses the keywords. However, you can write custom code that takes advantage of the keywords.
Display	Full Page View	Display to use for the portlet when you navigate to the portlet itself where My webMethods Server displays a page that contains only the portlet.
	Portlet View	Display to use for the portlet when you navigate to the page that contains the portlet.
Maintenance	Owner	The user name of the user that added the portlet to the page.
	Created On	The date and time the portlet was added to the page.
	Modified On	The date and time the portlet was last updated.
	Aliases	Optional aliases assigned to the portlet. For more information, see “Managing Portlet Aliases” on page 396 .

Preferences Tab

The **Preferences** tab contains properties that are specific to the portlet and that usually define the information that My webMethods Server displays in the portlet. For example, for the **HTML Text** tool, you can specify the text to display in the portlet on the **Preferences** tab.

To view the **Preferences** tab, you must view the properties from page editing mode. Not all portlets use a **Preferences** tab.

Layout Tab

The **Layout** tab contains properties that dictate how My webMethods Server is to display the portlet. To view the **Layout** tab, you must view the properties from page editing mode.

The following table lists the typical properties that are included on the **Layout** tab.

Section	Property	Description
Size and Positioning	Width	How wide to make the portlet. Specify the number of pixels to use for the portlet width.
	Height	How tall to make the portlet. Specify the number of pixels to use for the portlet height.
	Auto Positioned	How My webMethods Server positions the portlet when rendering the page. If Auto Position is toggled:

Section	Property	Description
		<ul style="list-style-type: none"> ■ On: My webMethods Server automatically positions the portlet, ignoring any positioning information stored with the portlet. ■ Off: My webMethods Server uses positioning information stored with the portlet to determine where to place the portlet on the page. As a result, My webMethods Server uses the location from when the page was last saved.
Display	Titlebar	<p>Whether you want My webMethods Server to display the portlet title bar. Select the check box to display the title bar; clear the check box to hide the title bar.</p> <p>Note: If you hide the title bar, when editing the page, select View As Expert to temporarily view the title bar so that you can manipulate the portlet on the page.</p>
	Border	<p>Whether you want My webMethods Server to display the portlet border. Select the check box to display the border; clear the check box to hide the border.</p> <p>Note: If you hide the border, when editing the page, select View As Expert to temporarily view the border to help you as you position and/or resize the portlet.</p>
	Minimized	<p>Whether you want My webMethods Server to initially display the portlet as minimized when it displays the page that contains the portlet. Users can restore the portlet to view its contents.</p>
	CSS Class	<p>A CSS class to apply to the portlet. Type the name of the class, omitting the leading period. For example, type nav for the “.nav” class.</p> <p>Specify a CSS class defined by a CSS style sheet included in the page, either the CSS style sheet that is:</p> <ul style="list-style-type: none"> ■ Used by the current skin ■ Included with a custom portlet in the page content or the current shell
	CSS Style	<p>A style to apply to the portlet. Type any style that is valid for use in a CSS file. For example, if you type border: 1pt dashed red, a dashed red line appears as a portlet border.</p>

Section	Property	Description
	Skin Background Image	<p>A background image to use for the portlet. Type the name of the image file from the current skin.</p> <p>You specify the name of the skin property. For example, to use the main logo image from the skin, type <code>images/logo.gif</code>. Determine the skin property name for an image by accessing the skin editor (via the Administration Dashboard > User Interface > Skin Administration portlet) and locating the image in the Images page. The skin property is "images/" followed by the property name, such as "images/logo.gif".</p> <div style="background-color: #f0f0f0; padding: 10px;"> <p>Note: To apply an image that is not in the current skin, specify the standard CSS background-image property in the CSS Style field. You can also specify properties in the CSS Style field to control how the background is displayed, such as whether it repeats, is centered, scrolls with the page, etc.</p> </div>

Metadata Tab

The **Metadata** tab contains properties that are specific to the portlet and that usually define the information that My webMethods Server displays in the portlet.

To view the **Metadata** tab, you must view the properties from page editing mode. Not all portlets use a **Preferences** tab.

Wiring Tab

Use the **Wiring** tab to wire the values of the properties. To view the **Wiring** tab, you must view the properties from page editing mode. For more information about wiring properties, see [“Wiring the Property of One Portlet to the Property of Another” on page 397](#) and [“Wiring a Principal Attribute to a Portlet Property” on page 398](#).

Modifying Portlet Properties

Configure portlet properties to specify display settings for the portlet and define how the portlet functions. You configure the properties for a portlet independently from the properties of the page on which the portlet resides.

» To modify portlet properties

1. As system administrator, navigate to and open the page containing the portlets you want to update.

2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. If the title bar of the portlet you want to update is hidden, select the **View As Expert** check box so that you can see the title bar.
4. In the portlet title bar, click , and then **Properties**.
5. Make the changes you want to the properties. For information about the properties, see [“Portlet Properties” on page 392](#).
6. At the bottom of the portlet, click **Apply**.
7. Click **Save**.

Tip:

If the title bar of the portlet you want to update is visible, you can update the portlet's property without switching to page editing mode. To view a portlet's properties while in regular view mode, in the portlet title bar click , and then **Properties**.

Managing Portlet Aliases

You can assign aliases to individual portlets. An alias is a new or simpler name for a portlet. By assigning an alias to a portlet, My webMethods Server recognizes the alias when it appears in a URL and automatically redirects a user to the portlet.

Use the aliases to access the portlets directly. Aliases are also useful to page developers when building multi-portlet applications; developers can use the aliases to allow one portlet to communicate with another.

➤ To add, edit or remove portlet aliases

1. As system administrator, navigate to and open the page containing the portlets you want to update.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. If the title bar of the portlet you want to update is hidden, select the **View As Expert** check box so that you can see the title bar.
4. In the portlet title bar, click , and then **Properties**
5. To add an alias:
 - a. In the **Aliases** section of the Properties page, click **Add**.

- b. Type the portlet alias that you want to add in the text box.
 - c. Click **OK**.
6. To change an existing alias:
 - a. In the **Aliases** section of the Properties page, select the alias you want to update and click **Edit**.
 - b. Type the new portlet alias in the text box.
 - c. Click **OK**.
7. To remove an alias, in the **Aliases** section of the Properties page, select the alias you want to remove and click **Remove**.
8. At the bottom of the page, click **Apply**.
9. Click **Save**.

Tip:

If the title bar of the portlet you want to update is visible, you can work with portlet aliases without switching to page editing mode. To view a portlet's properties while in regular view mode, in the portlet title bar click , and then click **Properties**.

Wiring the Property of One Portlet to the Property of Another

You can connect, or *wire*, any property of any portlet on a page to any property of any other portlet. When you wire one property to another, whenever the page is rendered, the server automatically sets the value of the destination property to the value of the source property. This feature allows you to quickly create a composite application out of several different portlets.

For example, if you create a page with two portlets, one a search form and one a search results display, you can wire the search form value (the source value) to the results display input value (the destination value). When a user enters some information into the search form and submits it, the server updates the results display to the results of that search.

➤ To wire one portlet to another

1. As system administrator, navigate to and open the page containing the portlet you want to wire.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. Decide which portlet is the wiring source and which is the wiring destination.

The destination portlet property receives its property value from the source portlet property.

4. If the title bar of the *destination portlet* is hidden, select the **View As Expert** check box so that you can see the title bar.
5. In the portlet title bar of the *destination portlet*, click , and then **Properties**.
6. Click the **Wiring** tab.

The wiring tab displays a list of properties on the destination portlet that are available for wiring.

7. For a target property that you want to wire, in the **SOURCE PORTLET** column, select the portlet that you want to use as the source.
8. In the **SOURCE PROPERTY** column, specify the specific property of the source portlet that you want to use.
 - a. Click **Browse**.
 - b. Select the property you want to use from the list.
 - c. Click **Select**.
9. Click **Apply**.
10. Click **Save**.

The run-time view of the page should display the destination portlet with whatever values are configured in the source portlets.

Tip:

If the title bar of the *destination portlet* is visible, you can wire the portlet without switching to page editing mode. To view the wiring page for a portlet, in the portlet title bar click , and then click **Wiring**.

Wiring a Principal Attribute to a Portlet Property

Users and groups have a set of Principal Attributes that you can wire to a portlet. For example, suppose a portlet uses a postal code to display certain information when a user views a page. If the postal code is provided by wiring from a Principal Attribute Provider, when the postal code attribute is modified within a directory service, the portlet uses the modified attribute value.

> To wire a Principal Attribute to a portlet

1. As system administrator, navigate to and open the page containing the portlet you want to wire.
2. In the page title bar, click , and then **Edit Page** to switch to page editing mode.
3. If the title bar of the portlet is hidden, select the **View As Expert** check box so that you can see the title bar.
4. In the portlet title bar of the portlet, click , and then **Properties**.
5. Click the **Wiring** tab.
6. For a target property that you want to wire, in the **SOURCE PORTLET** column, select **Other**.
The Select Portlet Resource window opens.
7. Click **Global Wiring Data**.
8. Move **Use Profile Wiring** to the **Selected Items** box and click **Select**.
The Select Portlet Resource window closes.
9. In the **SOURCE PROPERTY** column, specify the specific property to use for the source.
 - a. Click **Browse**.
The Select Wired Property window opens.
 - b. Select the property you want to use from the list.
 - c. Click **Select**.
10. Click **Apply**.
11. Click **Save**.

The portlet is now wired to use the attribute value belonging to the user who viewing the page on which the portlet resides.

Tip:

If the title bar of the portlet is visible, you can wire the portlet without switching to page editing mode. To view the wiring page for a portlet, in the portlet title bar click , and then click **Wiring**.

About Customizing the My webMethods Navigation

To customize the My webMethods navigation panel, you can:

- Add selections to the **Applications** section of the navigation panel. You can add:

- Additional pages to the **Monitoring** or **Administration** subsections
- New folders of pages to the **Applications** section

The folder becomes a new subsection within the **Applications** section. The items contained in the folder become selections within the new subsection.

For more information, see [“Adding Selections to the My webMethods Navigation” on page 400](#).

- Remove selections from the **Monitoring** or **Administration** subsections of the **Applications** section of the navigation panel. For more information, see [“Removing Selections from the My webMethods Navigation” on page 401](#).
- Hide standard tabs and sections of the navigation panel.

You can hide the **Navigate** tab completely. If you show the **Navigate** tab, you can hide either the **Applications** and/or **Workspaces** sections that are on the **Navigate** tab. You can also hide the **Tools** tab completely. For more information, see [“Hiding Standard Tabs and Sections of the My webMethods Navigation” on page 402](#).

- Completely replace the **Applications** section of the My webMethods navigation with a custom taxonomy. For more information, see [“Replacing the My webMethods Application Navigation with Your Own Taxonomy” on page 403](#).

Adding Selections to the My webMethods Navigation

System administrators can add custom pages or folders of pages to the **Applications** section of the My webMethods navigation.

➤ To add selections to the **Applications** section of the My webMethods navigation

1. Create the custom folder or custom page that you want to add to the navigation panel. For more information, see [“About Custom Folders and Pages” on page 382](#) and [“Creating Custom Pages” on page 383](#).

When creating the folder or page, save it in one of the following locations:

- To add a page to the **Monitoring** subsection of the **Applications** section, add it to:
Folders > My webMethods Applications > Fabric Tasks > Monitoring
- To add a page to the **Administration** subsection of the **Applications** section, add it to:
Folders > My webMethods Applications > Fabric Tasks > Administration
- To add a folder to create a new, custom subsection within the **Applications** section, add it to:
Folders > My webMethods Applications > Fabric Tasks

- To add a page to a custom subsection, add the page to the folder you are using for the subsection. For example, if you add a folder named “Custom Pages”, then you can add a page to the following location:

Folders > My webMethods Applications > Fabric Tasks > Custom Pages

2. As system administrator, navigate to and open the folder or page you want to add to the navigation panel.
3. In the page title bar, click  **Properties**.
4. Select the **Is Task Folder** check box to indicate that you want My webMethods Server to display the page in the My webMethods navigation panel.
5. Select the **Is Openable** check box if you want My webMethods Server to automatically open a page in a new tab when a user navigates to it from the navigation panel.
6. Click **Apply** to close the Properties page.

You can set Access Privileges for the page in the same manner you set Access Privileges for any other My webMethods page.

Removing Selections from the My webMethods Navigation

System administrators can remove both custom pages and folders, as well as, out-of-the box pages from the **Applications** section of the My webMethods navigation.

- To remove custom pages and/or folders, you can:
 - Permanently remove them by deleting them from their location within **Folders > My webMethods Applications > Fabric Tasks**
 - Temporarily hide them so that My webMethods does not display them in the navigation, by clearing the **Is Task Folder** check box in the page or folder’s properties.
- To remove out-of-the-box pages, it is recommended that you clear the **Is Task Folder** check box in the page’s properties rather than deleting the pages.
- To remove either one or both of the out-of-the-box **Monitoring** or **Administration** subsections, update the properties of **Folders > System > Shell Sections > Noodle Shell LeftNav > Leftnav**. For more information, see [“Hiding Standard Tabs and Sections of the My webMethods Navigation” on page 402](#).

➤ **To remove selections from the Applications section of the My webMethods navigation**

1. As system administrator, navigate to and open **Folders > My webMethods Applications > Fabric Tasks**.

2. To permanently remove a custom folder or page, delete it by clicking  in the title bar of the folder or page.
3. To hide a custom folder, custom page, or out-of-the-box page:
 - a. Open the properties for the folder or page by selecting , and then **Properties** in the title bar of the folder or page.
 - b. Clear the **Is Task Folder** check box.
 - c. Click **Apply**.

Hiding Standard Tabs and Sections of the My webMethods Navigation

System administrators can configure the properties of the left navigation page that is used for My webMethods navigation to:

- Completely hide the **Navigate** tab or **Tools** tab
- Hide either or both of the **Applications** section or **Workspaces** section that are displayed on the **Navigate** tab.

➤ To hide standard tabs and/or sections of the My webMethods navigation

1. As system administrator, navigate to the following location:
Folders > System > Shell Sections > Noodle Shell Leftnav > Leftnav
2. In the row for the Leftnav page, click , and then **Properties**.
3. In the **Portlet Preferences** section of the page, select or clear the following check boxes to show or hide tabs and sections:

The following table lists the display options, available on the Portlet Preferences section:

Check box	Description
Show Workspaces	Hides or shows the Workspaces section of the Navigate tab.
Show Applications	Hides or shows the Applications section of the Navigate tab.
Show Tools	Hides or shows the Tools tab.
Show Navigate	Hides or shows the Navigate tab.

4. Click **Apply**.

Replacing the My webMethods Application Navigation with Your Own Taxonomy

System administrators can completely replace the **Applications** section of the My webMethods navigation panel with a custom taxonomy.

To create a custom taxonomy, you create an alternative applications root page that contains the taxonomy you want to use. Then you configure the properties for the Leftnav page that My webMethods uses for its navigation panel to point to your new application root.

➤ To replace the My webMethods **Applications** taxonomy with a custom taxonomy

1. As system administrator, create a new page to use as the application root. It is recommended that you create the page within either of the following locations.

- Folders
- **Folders > Public Folders**

When creating the page, assign it a name that you want to appear in the navigation panel. The name you assign will be the new name of the **Applications** section.

2. Assign an alias to the new application root page.
 - a. In the row for the new application root page, select , and then **Properties**.
 - b. In the **Aliases** field, click **Add**.
 - c. In the text box, assign the alias you want to give the new application root and click **OK**.
 - d. Click **Apply**.
3. Build the taxonomy within the new applications root page.
 - To add a subsection to the taxonomy, add a page or folder to use as a container. Assign it a name that you want to use for the name of the subsection within the taxonomy.
 - To add a page that displays information, add a page. Assign it a name that you want to appear in the taxonomy.

For more information about creating custom folders and pages, see [“About Custom Folders and Pages” on page 382](#) and [“Creating Custom Pages” on page 383](#).
4. Configure the properties of the Leftnav page that My webMethods Server uses for the My webMethods navigation to point it to the new, custom application root.

- a. Navigate to the following location:

Folders > System > Shell Sections > Noodle Shell Leftnav > Leftnav

- b. In the row for the Leftnav page, select , and then **Properties**.
- c. In the **Applications Root** field, type the name of the alias you assigned to the applications root page you created.
- d. Click **Apply**.

Modifying the Bean Expiration Policy

By default, when a My webMethods user returns to a previous folder, such as a Task Inbox, the folder refreshes itself and unsaved changes to the folder are lost. System administrators can modify the bean expiration policy for a Fabric folder or a workspace template so it displays existing data when users return to the tab that contains it. Once set, the policy applies any time a user displays that folder or workspace in My webMethods.

> To modify bean expiration policy

1. As system administrator do one of the following:
 - For Fabric folders navigate to **Folders > My webMethods Applications > Fabric Tasks**.
Folders that allow the you to modify the bean expiration policy are identified by the  icon.
 - For workspace templates navigate to **Folders > System > Templates > Workspace Templates > Default Workspace Template**.
2. In the Fabric folder title bar or in the Default Workspace Template, click  **Tools > Properties**.
3. In the **beanExpirePolicy** field, type `do_not_expire`.
4. Click **Apply**.

Make this change individually for each folder you want to modify.

About Customizing the My webMethods Look-And-Feel

System administrators can change the My webMethods look-and-feel by updating the skin that My webMethods uses. To customize the My webMethods look-and-feel, you can:

- Change the Software AG logo, for example, to display your own corporate logo.

- Change the colors that are used in the My webMethods user interface.
- Completely customize the My webMethods look-and-feel by using a custom shell and skin.

Replacing the Logo in the My webMethods User Interface

System administrators can change the images used in the My webMethods user interface by updating the images in the skin that My webMethods uses. To change the Software AG logo that appears at the top of the interface, update the logo.gif image. For more information about skins, see [“Customizing Skins” on page 433](#).

My webMethods uses the “Noodle - Twilight” skin. It is recommended that you do not update the “Noodle - Twilight” skin, but rather make a copy of it, update the copy, then configure the My webMethods to use the modified copy of the skin.

➤ To replace the logo in the My webMethods user interface

1. Make a copy of the “Noodle - Twilight” skin.
 - a. As system administrator, navigate to **Folders > Administrative Folders > Administration Dashboard > User Interface > Skin Administration**.
 - b. Click the **Create New Skin** link at the top of the page.
 - c. On the **Create New Skin** page, specify the following:

The following table lists the information you must provide when creating new skins:

Field	Description
System Name	A short name that contains only letters, numbers, and the underscore character. My webMethods Server uses this name internally.
Display Name	A descriptive name that My webMethods Server uses when it displays the skin name in the user interface.
Parent Skin	Noodle - Twilight

- d. Click **Save**.
2. Edit the properties for the new skin by selecting , and then **Edit**.
3. Click the **Images** tab.
4. Update the image for the logo.gif image. For instructions, see [“Replacing Images in a Skin” on page 439](#).

5. Configure My webMethods so that it uses the updated skin.
 - a. Navigate to **Folders > Administrative Folders > Administration Dashboard > User Interface > Manage Skin Rules**.
 - b. Click the **My webMethods** link to open the skin rule that My webMethods uses.
 - c. On the **Modify Rules** tab, in the **Results** list, select the new skin.
 - d. Click **Update Rule**.

Changing the Color Scheme of the My webMethods User Interface

System administrators can change the colors used in the My webMethods user interface by updating the colors in the skin that My webMethods uses. For more information about skins, see [“Customizing Skins” on page 433](#).

My webMethods uses the “Noodle - Twilight” skin. It is recommended that you do not update the “Noodle - Twilight” skin, but rather make a copy of it, update the copy, then configure the My webMethods to use the modified copy of the skin.

➤ To change the colors in the My webMethods user interface

1. Make a copy of the “Noodle - Twilight” skin.
 - a. As system administrator, navigate to **Folders > Administrative Folders > Administration Dashboard > User Interface > Skin Administration**.
 - b. Click the **Create New Skin** link, which is at the top of the page.
 - c. Complete the fields for the new skin.

The following table lists the required fields to configure when creating a new skin for My webMethods:

Field	Description
System Name	A short name that contains only letters, numbers, and the underscore character. My webMethods Server uses this name internally.
Display Name	A descriptive name that My webMethods Server uses when displaying the skin name in the user interface.
Parent Skin	Noodle - Twilight

- d. Click **Save**.
2. Edit the properties for the new skin by clicking , and then **Edit**.
3. Click the **Colors** tab.
4. Update the colors. For instructions, see [“Replacing Colors Using a Color Picker”](#) on page 441.
5. Configure My webMethods so that it uses the updated skin.
 - a. Navigate to **Folders > Administrative Folders > Administration Dashboard > User Interface > Manage Skin Rules**.
 - b. Click the **My webMethods** link to open the skin rule that My webMethods uses.
 - c. On the **Modify Rules** tab, in the **Results** list, select the new skin.
 - d. Click **Update Rule**.

Applying a Custom Skin and Shell to My webMethods

System administrators can completely change the look-and-feel of My webMethods by configuring it to use a custom skin and shell.

➤ To apply a custom skin and shell to My webMethods

1. Create the custom shell you want to use. For instructions, see [“Working with Shells in My webMethods Server”](#) on page 459.
2. Create the custom skin you want to use. For instructions, see [“Customizing Skins”](#) on page 433.
3. Configure My webMethods so that it uses your custom shell:
 - a. Navigate to **Folders > Administrative Folders > Administration Dashboard > User Interface > Manage Shell Rules**.
 - b. Click the **My webMethods** link to open the shell rule that My webMethods uses.
 - c. On the **Modify Rules** tab, in the **Results** list, select the name of your custom shell.
 - d. Click **Update Rule**.
4. Configure My webMethods so that it uses your custom skin:

- a. Navigate to **Folders > Administrative Folders > Administration Dashboard > User Interface > Manage Skin Rules**.
- b. Click the **My webMethods** link to open the skin rule that My webMethods uses.
- c. On the **Modify Rules** tab, in the **Results** list, select the name of your custom skin.
- d. Click **Update Rule**.

Building a Simple Front-End Page to My webMethods

If you have users that need access to only a few My webMethods pages and you want to simplify their access to those pages, you can build a simple page that provides links to only the pages the users require.

The links you include in the front-end page can use URLs that use a page alias. Some My webMethods application pages are assigned aliases out-of-the-box. If an application page that you want to use does not have an alias, you can assign one to the page. After an alias is assigned, as an alternative to selecting the page from the My webMethods navigation panel, you can navigate directly to it by entering a URL that uses the page alias.

> To build a simple front-end page to My webMethods

1. Determine the application pages that you want to include in the front-end page.
2. For each application page, determine its page an alias or assign one if the page does not have an alias.
 - a. As system administrator, navigate to the following:
Folders > My webMethods Applications > Fabric Task
 - b. Further navigate to and open the Application page you want.
 - c. In the page title bar, click , and then **Properties**
 - d. In the **Aliases** section of the screen, note the page alias. If there is no page alias, assign one.
 - e. Click **Apply** to close the Properties page.
3. Build a webpage that includes links to the Application pages. Use the following for the URL where:
 - *host_name* and *port_number* are the host name and port number of the My webMethods Server

- *alias* is the alias name of the application page

`http://host_name:port_number/alias`

For example if the host name and port of My webMethods Server is “mws.company.com:8585” and you want to access the Tasks Inbox page, which has the alias “webm.apps.workflow.inbox”, use the following URL:

`http://mws.company.com:8585/webm.apps.workflow.inbox`

Creating Links for Single Sign-On

Single sign-on is the ability for a user to log into one application and then use other applications without having to log into each one separately. My webMethods Server supports single sign-on through the Security Assertion Markup Language (SAML), an XML-based framework for the exchange of security information.

To take advantage of single sign-on, a user must be known on both the source server and the target entity. In most cases, common knowledge of a user is provided by use of the same directory service. For more information on configuring a server to be used as a target for single sign-on, see [“Configuring My webMethods Server Single Sign-On” on page 277](#).

On any page, you can add a link to a SAML target entity, such as a server. If the target accepts SAML assertions from the source server, when a known user clicks the link, no login credentials are required. If the target entity does not accept SAML assertions from the source server, or if the user is not known on the target entity, login credentials may be required.

Under the SAML specification, an intermediary called an artifact receiver can perform authentication on behalf of the target web application. In such a case, the SAML source requires two URLs: one for the Artifact Receiver and one for the target web application.

You can place one or more SAML links on any page you have permission to edit.

➤ To create a SAML link on a source page

1. In the upper right-hand corner of the page, click , and then **Edit Page**.
2. In the **Root** list of the **Available Portlets** panel, click **Links**.
3. In the **Links** list of the **Available Portlets** panel, drag the **Single Sign-on Link** portlet and drop it onto the page at the location where you want to add the link.

A red box appears beneath the cursor location whenever the cursor is over a valid page location, indicating where the portlet would be positioned if you released the mouse button.

4. On the left side of the page control area, click **Save**.
5. At the right edge of the title bar for the single sign-on portlet, click , and then **Properties**.

6. In the **Properties** specify:

The following table lists the properties that you specify to configure the Single Sign-On Link portlet:

Property	Description
Name	Replace <code>Single Sign-on Link</code> with the text that is to go with the link.
SAML Authentication URL	Type the URL for a resource on the target computer. The target can be any page on a server. If you are connecting to a web application through a SAML Artifact Receiver, use this field for the Artifact Receiver URL.
Use POST or GET	Determines the method used to pass data to the target computer. POST Passes data to a gateway program's STDIN. POST, the default, is the preferred method for single sign-on data. GET Passes data as a string appended to the URL after a question mark.
Artifact Parameter Name	If this is a SAML connection with another server or other webMethods product, do not change the default value <code>SAMLart</code> . If this is a SAML connection to a third-party source, type the artifact parameter name used by the third-party application.
Application Target URL	If you have typed the URL for a SAML Artifact Receiver in the SAML Authentication URL field, type the URL for a web application. Otherwise, leave this field empty.

7. Click **Apply**.

23 Managing Workspaces in My webMethods Server

- [About Workspaces](#) 412
- [Administration Tasks for Workspaces](#) 412
- [Expert User Features for Workspace Development](#) 416
- [Workspace Actions You Can Perform from the Workspace Management Page](#) 422
- [About the My webMethods Tools Navigation](#) 429

About Workspaces

Workspaces are pages that users create and use as work areas for some specific purpose. Users build the content of the workspace by dragging portlets on to the workspace. For example, users might create a workspace to gather information about an issue they need to solve. They might attach files to the workspace related to the issue by adding the Attachments tool to the workspace. If they have a screen shot that illustrates the issue, they might add the Image tool to the workspace.

Users can also share workspaces with other users so that multiple users can work together. For example, a user might add the Attachments tool to the workspace so that the users sharing the workspace can upload and share files. The owner of the workspace sets the permissions that dictate that actions that other users sharing the workspace can perform.

Users can create as many workspaces as they need. When the workspace is no longer needed, for example because the issue for which it was established is resolved, users can simply delete the workspace.

For more information about the basic use of workspaces, such as, creating workspaces, sharing workspaces, and adding portlets to workspaces, see *Working with My webMethods*.

System administrators and My webMethods administrators can restrict or enhance the a user's workspace functionality. For more information, see [“Administration Tasks for Workspaces” on page 412](#).

- To restrict functionality, an administrator can set users' permissions to deny functionality that they are granted out-of-the-box. For example, you can deny the functional privilege that allows a user to create new workspaces.
- To enhance functionality, an administrator can set users' permissions to grant functionality that they are *not* granted out-of-the-box. For example, you can grant:
 - The functional privilege that makes a user an expert user. Expert users have access to additional properties and menu actions that aid in developing workspaces. For more information, see [“Expert User Features for Workspace Development” on page 416](#).
 - Access to the Navigate > Applications > Administration > System-Wide > Workspace Management page from which a user can perform actions against workspaces. For more information, see [“Workspace Actions You Can Perform from the Workspace Management Page” on page 422](#).

Additionally, system administrators can customize the **Tools** tab of the My webMethods navigation. For more information, see [“About the My webMethods Tools Navigation” on page 429](#).

Administration Tasks for Workspaces

System administrators and My webMethods administrators can enhance or restrict the a user's workspace functionality by performing the following tasks:

- Grant users access to the [Navigate > Applications > Administration > System-Wide > Workspace Management](#) page, which allows users to search for and take action on workspaces. For more information, see [“Allowing Users to Access the Workspace Management Page”](#) on page 413.
- Prohibit users from performing basic workspace actions, such as viewing workspaces and creating workspaces. For more information, see [“Workspace Functional Privileges”](#) on page 414 and [“Controlling the Workspace Functions a User Can Perform”](#) on page 415.
- Grant users Expert Workspace Development privileges, giving them additional functionality that they can use when building workspaces. For more information, see [“Workspace Functional Privileges”](#) on page 414, [“Controlling the Workspace Functions a User Can Perform”](#) on page 415, and [“Expert User Features for Workspace Development”](#) on page 416.
- Remove the **Workspaces** section of the My webMethods navigation for all users. For more information, see [“Hiding Standard Tabs and Sections of the My webMethods Navigation”](#) on page 402.
- Customize the taxonomy displayed on the **Tools** tab of the My webMethods navigation. For more information, see [“Customizing the My webMethods Workspace Tools”](#) on page 430.

If a system administrator needs to take action to fix or delete a user’s workspace and cannot do so via the [Navigate > Applications > Administration > System-Wide > Workspace Management](#) page, they can attempt to correct the problem by accessing it in the user’s personal folder. My webMethods Server stores each workspace in the personal folder of the user who created the workspace. For example, if the user “jsmith” creates a workspace, you can use the system administrator user interface to navigate to the workspace in [Folders > Users > jsmith's Root Folder > Workspaces](#).

Allowing Users to Access the Workspace Management Page

My webMethods includes the [Navigate > Applications > Administration > System-Wide > Workspace Management](#) page that allows users to take actions against workspaces. For a description, see [“Workspace Actions You Can Perform from the Workspace Management Page”](#) on page 422.

By default, only My webMethods administrators have access to the Workspace Management page; end users do not. However, end users can perform many of the same actions from the right-click menu in the **Workspaces** section of the My webMethods navigation and from the menu on the tab of an open workspace. The workspace actions that are only available via the Workspace Management page are:

- Searching for workspaces.
- Exporting workspaces to a file.
- Importing workspaces that were previously exported to a file.

System administrators and My webMethods administrators can assign permissions to allow users, groups, and/or roles to access the Workspace Management page.

➤ **To allow users access the Workspace Management page**

1. Navigate to the Permissions Management page.
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > Permissions Management.**
 - As system administrator: **Folders > Administrative Folders > Administration Dashboard > Configuration > Permissions Management.**
2. Use the Permissions Management page to select the users, groups, and/or roles to which you want to grant access and to set the access privilege to the Workspace Management Page.

When setting privileges, in the **Permissions** tree select the following:

Access Privileges > Administration > System-Wide > Workspace Management

For instructions for how to use the Permissions Management page to set access privileges, see [“Managing Access Privileges and Functional Privileges” on page 173.](#)

Workspace Functional Privileges

Workspace functional privileges govern the actions that users can take against workspaces. System administrators and My webMethods administrators can change the default permissions for users, groups, and/or roles. For more information, see [“Controlling the Workspace Functions a User Can Perform” on page 415.](#)

The following table lists the workspace functional privileges, available in My webMethods and their default settings:

Functional Privilege	Default Setting	Description
View Workspaces	Granted	<p>Controls whether users can view workspaces.</p> <p>My webMethods Server does not display the Workspaces section on the Navigate tab of the My webMethods navigation when a user is denied the View Workspaces functional privilege.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: Even when the View Workspaces functional privilege is denied, if users have access to the Workspace Management page, they can search for and open workspaces. For more information, see “Allowing Users to Access the Workspace Management Page” on page 413 and “Opening a Workspace” on page 424.</p> </div>
Edit Workspaces	Granted	<p>Controls whether users can add portlets to a workspace.</p> <p>My webMethods Server does not display the Tools tab of the My webMethods navigation when a user is denied the Edit Workspaces functional privilege.</p>

Functional Privilege	Default Setting	Description
		<p>Note: Even when the Edit Workspaces functional privilege is denied, users can still set workspace properties, reposition portlets in a workspace, set portlet properties, and delete portlets from workspaces.</p>
Create Workspaces	Granted	<p>Controls whether users can create new workspaces.</p> <p>My webMethods Server does not display the New tab in My webMethods when a user is denied the Create Workspaces functional privilege.</p>
Import Workspaces	Granted	<p>Controls whether users can import workspaces that were previously exported to a file.</p> <p>My webMethods Server does not display the Import button on the Workspace Management page when a user is denied the Import Workspaces functional privilege.</p>
Expert Workspace Development	Denied	<p>Provides users with extra functionality that they can use when building workspaces and pages. Users with Expert Workspace Development functional privilege have access to additional properties and menu actions. For more information, see “Expert User Features for Workspace Development” on page 416.</p>

Controlling the Workspace Functions a User Can Perform

System administrators and My webMethods administrators can use the following procedure to grant or deny users, groups, and/or roles the functional privileges described in [“Workspace Functional Privileges” on page 414.](#)

➤ To grant or deny users workspace functional privileges

- Navigate to the Permissions Management page.
 - In My webMethods: **Navigate > Applications > Administration > System-Wide > Permissions Management.**
 - As system administrator: **Folders > Administrative Folders > Administration Dashboard > Configuration > Permissions Management.**
- Use the Permissions Management page to select the users, groups, and/or roles with which you want to work and to grant or deny the functional privileges.

To grant or deny workspace functional privileges, navigate to the following part of the **Permissions** tree and grant or deny the privileges described in [“Workspace Functional Privileges” on page 414](#).

Functional Privileges > General

For instructions for how to use the Permissions Management page to set functional privileges, see [“Managing Access Privileges and Functional Privileges” on page 173](#).

Expert User Features for Workspace Development

Expert users are users who have been assigned the Expert Workspace Development functional privilege. For instructions on how administrators can assign this functional privilege, see [“Controlling the Workspace Functions a User Can Perform” on page 415](#).

Expert users have access to additional:

- **Workspace properties** to better manage the layout of a workspace. For more information, see [“Workspace Properties for Expert Users” on page 416](#).
- **Properties for portlets in a workspace** to better manage the content of portlets and how they appear on the workspace. For example, expert users can set up a portlet so that its title bar does not display. For more information, see [“Portlet Properties for Expert Users” on page 419](#).
- **Portlet menu actions** to:
 - Indicate that you want My webMethods Server to position a portlet when rendering the workspace
 - Set permissions for portlets

For more information, see [“Portlet Menu Options for Expert Users” on page 421](#).

Note:

Expert users also see these additional properties and portlet menu options when working with regular pages, not just workspaces.

Workspace Properties for Expert Users

General Tab

The workspace properties on the **General** tab are identical for users with and without the Expert Workspace Development privilege. For information about these workspace properties, see *Working with My webMethods*.

Layout Tab

The **Layout** tab is available for both users with and without the Expert Workspace Development privilege, but some of the properties are only available for expert users.

The following table describes the workspace properties on the **Layout** tab:

Property	Available for	Description
View As	expert users	<p>This property indicates whether you want to view column, row, and portlet borders and hidden portlet title bars.</p> <ul style="list-style-type: none"> ■ Select End User if you do <i>not</i> want to view borders or hidden title bars. ■ Select Expert User if you want to view borders and hidden title bars. <p>Borders are helpful when you are positioning portlets. Also, if you set a portlet's properties so that the title bar is hidden, you must view the page as an expert user to re-display the title bar so that you can take action on that portlet, for example, to move it or access its properties.</p>
Editable Canvas	expert users	<p>This property indicates whether you want users viewing the workspace to be able to reposition and/or resize portlets.</p> <p>Clear the Editable Canvas check box to prevent users from inadvertently changing the layout while using the workspace. If you select a free form layout for the workspace, a user can inadvertently change the page simply by clicking a portlet.</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: When the Editable Canvas check box is selected, users can change the layout. This is true even when users are denied permissions to edit a workspace, although they are prevented from saving those edits. However, when the Editable Canvas check box is cleared, all users are prevented from changing the layout.</p> </div>
Columns	all users	<p>This property specifies the layout to use for the workspace. The layout defines how the portlets within a workspace can be positioned. You can define either a:</p> <ul style="list-style-type: none"> ■ Column layout. In this layout, all the portlets in a workspace are aligned in columns. Portlets cannot overlap within a column. You can define a workspace to have one, two, three, or four columns.

Property	Available for	Description
		<ul style="list-style-type: none"> ■ Free Form layout. In this layout, you can place portlets on the workspace in any location. Portlets can overlap and even completely cover one another. This is the default for a new workspace.
Attributes	expert users	When using a column layout, this property specifies attributes for the column that you can use for your own use. Out-of-the-box, My webMethods Server does not provide functionality that uses the attributes. However, you can write custom code that takes advantage of the attributes.
Width	all users	When using a column layout, this property specifies the percentage of the workspace to use for the width of the column.
Word Wrap	all users	When using a column layout, this property specifies whether you want to wrap long text lines within portlets to fit the column size.
Horizontal Alignment	all users	When using a column layout, this property specifies how the portlets should be aligned horizontally in the column.
Vertical Alignment	all users	When using a column layout, this property specifies how the portlets should be aligned vertically in the column.
CSS Class	expert users	<p>When using a column layout, this property specifies a CSS class to apply to the column. Type the name of the class, omitting the leading period. For example, type <code>nav</code> for the <code>“.nav”</code> class.</p> <p>Specify a CSS class defined by a CSS style sheet included in the workspace, either the CSS style sheet that is:</p> <ul style="list-style-type: none"> ■ Used by the current skin ■ Included with a custom portlet in the workspace content or the current shell
CSS Style	expert users	When using a column layout, this property specifies a style to apply to the column. Type any style that is valid for use in a CSS file. For example, if you type <code>border: 1pt dashed red</code> , a dashed red line appears as a portlet border.

Property	Available for	Description
Skin Background Image	expert users	<p>When using a column layout, this property specifies an image from the current skin to use as the background image for the column.</p> <p>Type the name of the skin property. For example, to use the main logo image from the skin, type <code>images/logo.gif</code>. System administrators can determine the skin property name for an image by accessing the skin editor (via the Administration Dashboard > User Interface > Skin Administration portlet) and locating the image in the Images page. The skin property is "images/" followed by the property name, such as "images/logo.gif".</p> <div style="background-color: #f0f0f0; padding: 5px;"> <p>Note: To apply an image that is not in the current skin, specify the standard CSS background-image property in the CSS Style field. You can also specify properties in the CSS Style field to control how the background is displayed, such as whether it repeats, is centered, scrolls with the page, etc.</p> </div>

Portlet Properties for Expert Users

General Tab

The portlet properties on the **General** tab are identical for users with and without the Expert Workspace Development privilege. For information about these workspace properties, see *Working with My webMethods*.

Preferences Tab

The **Preferences** tab is available for both users with and without the Expert Workspace Development privilege. Not all portlets have a **Preferences** tab.

The properties on the **Preferences** tab are specific to each portlet and usually define the information that My webMethods Server displays in the portlet. For example, for the HTML Text tool you can specify the text to display in the portlet on the **Preferences** tab.

Layout Tab

The **Layout** tab is available for only users with Expert Workspace Development privilege. The following table describes the portlet properties on the **Layout** tab:

Property	Description
Attributes	Attributes for your own use. Out-of-the-box, My webMethods Server does not provide functionality that uses the attributes. However, you can write custom code that takes advantage of the attributes.
Width	How wide to make the portlet. Specify the number of pixels to use for the portlet width.
Height	How tall to make the portlet. Specify the number of pixels to use for the portlet height.
Titlebar	<p>Whether you want My webMethods Server to display the portlet title bar. Select the check box to display the title bar; clear the check box to hide the title bar.</p> <p>Note: If you hide the title bar, use the View As property, which is on the Layout tab of the workspace properties, to temporarily view the title bar so that you can manipulate the portlet on the workspace.</p>
Border	<p>Whether you want My webMethods Server to display the portlet border. Select the check box to display the border; clear the check box to hide the border.</p> <p>Note: If you hide the title bar, use the View As property, which is on the Layout tab of the workspace properties, to temporarily view the title bar so that you can manipulate the portlet on the workspace.</p>
Minimized	Whether you want My webMethods Server to initially display the portlet as minimized when it displays the workspace that contains the portlet. Users can restore the portlet to view its contents.
CSS Class	<p>A CSS class to apply to the portlet. Type the name of the class, omitting the leading period. For example, type <code>nav</code> for the <code>".nav"</code> class.</p> <p>Specify a CSS class defined by a CSS style sheet included in the workspace, either a CSS style sheet that is:</p> <ul style="list-style-type: none"> ■ Used by the current skin ■ Included with a custom portlet in the workspace content or the current shell
CSS Style	A style to apply to the portlet. Type any style that is valid for use in a CSS file. For example, if you type <code>border: 1pt dashed red</code> , a dashed red line appears as a portlet border
Skin Background Image	An image from the current skin to use as the background image for the portlet.

Property	Description
	<p>Type the name of the skin property. For example, to use the main logo image from the skin, type <code>images/logo.gif</code>. Expert users can ask system administrators for information about the properties available. System administrators can determine the skin property name for an image by accessing the skin editor (via the Administration Dashboard > User Interface > Skin Administration portlet) and locating the image in the Images page. The skin property is "images/" followed by the property name, such as "images/logo.gif".</p> <p>Note: To apply an image that is not in the current skin, specify the standard CSS background-image property in the CSS Style field. You can also specify properties in the CSS Style field to control how the background is displayed, such as whether it repeats, is centered, scrolls with the page, etc.</p>

Metadata Tab

The **Metadata** tab is available only users with Expert Workspace Development privilege. The properties on the **Metadata** tab are specific to each portlet and usually define the information that My webMethods Server displays in the portlet.

Portlet Menu Options for Expert Users

Expert users can access additional menu actions by clicking ▾ in the title bar of a portlet. The following table lists the portlet menu actions that are available only for users with Expert Workspace Development privilege.

Action	Description
Auto Position	<p>Defines how My webMethods Server positions the portlet when rendering the workspace. If Auto Position is toggled:</p> <ul style="list-style-type: none"> ■ On: My webMethods Server automatically positions the portlet, ignoring any positioning information stored with the portlet. ■ Off: My webMethods Server uses positioning information stored with the portlet to determine where to place the portlet on the workspace. As a result, My webMethods Server uses the location from when the workspace was last saved.
Permissions	<p>Sets permissions for how users can interact with the portlet on the page. For more information about the permissions and how to set them, see “Managing Permissions for an Individual Resource” on page 174 and “Using Security Realms” on page 176.</p>

Workspace Actions You Can Perform from the Workspace Management Page

Use the [Navigate > Applications > Administration > System-Wide > Workspace Management](#) page to perform actions against workspaces. By default, only My webMethods administrators have access to the Workspace Management page. My webMethods administrators can grant other users access to the Workspace Management page. For more information, see [“Allowing Users to Access the Workspace Management Page” on page 413](#).

Performing a Keyword Search for Workspaces

Use this search to find workspaces by specifying text found in the names, description, or keywords of the workspaces. If you want to specify detailed search criteria, see [“Performing an Advanced Search for Workspaces” on page 423](#).

You can find workspaces for which you are the owner or shared workspaces for which you have at least View Only permissions. My webMethods administrators can search for workspaces owned by any users.

Note:

For more information about sharing workspaces, see [“Sharing a Workspace” on page 426](#). For more information about workspace permissions, see *Working with My webMethods*.

➤ **To perform a keyword search for workspaces**

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Workspace Management**.
2. Click the **Keyword** search tab if it is not already displayed.

Tip:

For instructions for how to use the **Advanced** tab, see [“Performing an Advanced Search for Workspaces” on page 423](#). For instructions for how to use the **Saved** and **Options** tabs, see *Working with My webMethods*.

3. In the text box, type one or more strings that are contained in the names of the workspaces you want to find.

For more information about how to specify keywords and using special characters, see *Working with My webMethods*.

4. Click **Search**.

My webMethods Server displays the search results in the Results panel (below the Search panel).

Performing an Advanced Search for Workspaces

Use an advanced search to specify detailed criteria to search for workspaces.

You can find workspaces for which you are the owner or shared workspaces for which you have at least View Only permissions. My webMethods administrators can search for workspaces owned by any users.

Note:

For more information about sharing workspaces, see [“Sharing a Workspace” on page 426](#). For more information about workspace permissions, see *Working with My webMethods*.

➤ To perform an advanced search for workspaces

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Workspace Management**.
2. Click the **Advanced** search tab if it is not already displayed.

Tip:

For instructions for how to use the **Keyword** tab, see [“Performing a Keyword Search for Workspaces” on page 422](#). For instructions for how to use the **Saved** and **Options** tabs, see *Working with My webMethods*.

3. In the **Keywords** field, optionally type text found in the names, description, or keywords of the workspaces you want to find.
4. To specify detailed criteria, specify each criterion you want to use:
 - a. From the list, select a criterion you want to use:

The following table lists the available search criteria, and how to configure each criterion:

Criterion	Configuration
Name	In the text box, type a string contained in the names of the workspaces you want to find.
Description	In the text box, type a string contained in the descriptions of the workspaces you want to find.
Keyword	In the text box, type a string contained in the keywords of the workspaces you want to find.
Alias	In the text box, type a string contained in the aliases assigned to the workspaces that you want to use to find.
Type	Select the type of workspaces you want to find from the list.

Criterion	Configuration
Owner	<ol style="list-style-type: none">Click Browse.Use the Select Users window to search for and select the users whose workspaces you want to find. For instructions on how to search for users, see <i>Working with My webMethods</i>.Click Apply.
Created Date	In the Date Range list, select the date range you want to use to search for workspaces that were created within the date range you specify.
Modified Date	In the Date Range list, select the date range you want to use to search for workspaces that were modified within the date range you specify.

- To add another criterion, select  **Add** and repeat this step.
- In the **Search Condition** list, select:
 - **AND** if you want My webMethods Server to search for workspaces that meet *all* the criteria you specify.
 - **OR** if you want My webMethods Server to search for workspaces that meet *any* the criteria you specify.
 - Click **Search**.

Opening a Workspace

To view a workspace, you can open it in a new tab.

> To open a workspace

- Search for the workspaces that you want to open. For instructions, see [“Performing a Keyword Search for Workspaces” on page 422](#).
- In the search results in the row for a workspace you want to open, click  and select **Open In New Tab**.

Adding a Workspace to Your Navigation

You can add workspaces to the **Workspaces** section of your My webMethods navigation. After adding a workspace, you can open it by simply clicking on its name in the navigation.

When you add a workspace, you select the folder where you want the workspace to appear. By default, the **Workspaces** section of the navigation has two folders, **Recently Added** and **My**

Workspaces. You can add additional folders. For instructions for how to add additional workspace folders and also how to remove workspaces from the navigation, see *Working with My webMethods*.

➤ To add workspaces to your navigation

1. Search for the workspaces with which you want to work. For instructions, see [“Performing a Keyword Search for Workspaces” on page 422](#) or [“Performing an Advanced Search for Workspaces” on page 423](#).
2. You can add workspaces to your navigation by performing either of the following:
 - In the search results, select the check box beside each workspace that you want to add to your navigation, and click **Add to Navigation**.
 - In the row for a workspace you want to add to your navigation, click  and select **Add to Navigation**.
3. In the Add to Navigation window, select the folder to which you want to add the workspace.
4. Click **Apply**.

Deleting a Workspace

If you no longer need a workspace, you can delete it. If you want to save a backup of the workspace before deleting it, you can export it to a file first. For instructions, see [“Exporting Workspaces” on page 428](#).

➤ To delete workspaces

1. Search for the workspaces that you want to delete. For instructions, see [“Performing a Keyword Search for Workspaces” on page 422](#) or [“Performing an Advanced Search for Workspaces” on page 423](#).
2. If you want to notify users that share the workspace about the deletion, select the **Notify collaborators when a workspace is deleted** check box.
3. You can delete workspaces by performing either of the following:
 - In the search results, select the check box beside each workspace that you want to delete, and click **Delete**.
 - In the row for a workspace you want to delete, click  and select **Delete Workspace**.

Renaming a Workspace

Use the following procedure to rename a workspace.

> To rename a workspace

1. Search for the workspaces that you want to rename. For instructions, see [“Performing a Keyword Search for Workspaces” on page 422](#) or [“Performing an Advanced Search for Workspaces” on page 423](#).
2. In the search results in the row for a workspace you want to rename, click  and select **Rename Workspace**.
3. In the Rename Workspace window, type a new name in the **New Name** field.
4. Click **Apply**.

Sharing a Workspace

You can share workspaces with other users, groups, or roles. For details about sharing workspaces, including permissions you can assign when sharing a workspace, see *Working with My webMethods*.

> To share a workspace

1. Search for the workspaces that you want to share. For instructions, see [“Performing a Keyword Search for Workspaces” on page 422](#) or [“Performing an Advanced Search for Workspaces” on page 423](#).
2. In the search results in the row for a workspace you want to share, click  and select **Share Workspace**.
3. In the Workspace Sharing window click **Add**.
4. Use the Select Principal(s) window to search for and select the users, groups, and roles with whom you want to share the workspace. When you are done, click **Apply** to close the Select Principal(s) window. For more information about using the Select Principal(s) window to search for users, groups, and/or roles, see *Working with My webMethods*.

For each selected user, group, and/or role, My webMethods adds a row to the Workspace Sharing window.

5. For each user, group, and/or role that you selected, select the permissions you want to assign that user, group, or role from the list in the **Permissions** column. For more information about the permissions you can assign, see *Working with My webMethods*.
6. If you want to notify the users affected by this change, select the **Notify collaborators when the workspace is shared or unshared** check box.
7. Click **Apply**.

Unsharing Workspaces

Use the following procedure when you no longer want to share a workspace with a user, group, or role.

> To unshare a workspace

1. Search for the workspaces that you no longer want to share. For instructions, see [“Performing a Keyword Search for Workspaces” on page 422](#) or [“Performing an Advanced Search for Workspaces” on page 423](#).
2. In the search results in the row for a workspace you want to unshare, click  and select **Share Workspace**.
3. In the Workspace Sharing window, select the check box for the users, groups, and/or roles with which you no longer want to share access to the workspace.
4. If you want to notify the users affected by this change, select the **Notify collaborators when the workspace is shared or unshared** check box.
5. Click **Delete** and then **Apply**.

Setting the Properties of a Workspace

Use the following procedure to view and/or set the general properties for a workspace.

> To set workspace properties

1. Search for the workspaces with which you want to work. For instructions, see [“Performing a Keyword Search for Workspaces” on page 422](#) or [“Performing an Advanced Search for Workspaces” on page 423](#).
2. In the search results in the row for the workspace for which you want to set properties, click  and select **Properties**.
3. In the Workspace Properties window, update the general properties.
For a description of the general properties, see *Working with My webMethods*.
4. Click **Apply**.

Exporting Workspaces

You can export workspaces to save a copy of a workspace in a file. You can then import the workspace into the same or another My webMethods Server.

You can use the export/import functionality if you want to migrate a workspace from one My webMethods Server to another. You might also want to export a workspace to save a copy before deleting a workspace; if you decide you need it again in the future, you can then import it from the file.

➤ To export workspaces

1. Search for the workspaces you want to export. For instructions, see [“Performing a Keyword Search for Workspaces” on page 422](#) or [“Performing an Advanced Search for Workspaces” on page 423](#).
2. In the search results, select the check box beside each workspace that you want to export, and click **Export**.
3. In the Export Workspaces window, specify a file name for the file that will contain the exported workspaces.
4. If you also want to include information about how the workspaces are shared so that information is set when you later import the workspaces, select the **Include Shared Settings** check box.
5. Click **Apply**.
6. In the File Download window, click **Save**.
7. In the Save As window, select the location where you want to save the exported workspaces file and click **Save**.

Importing Workspaces

If you have previously exported workspaces to a file, you can import them into My webMethods Server.

My webMethods Server imports the workspace into the personal folder of the user that performed the import. It is possible for multiple users to import the same workspace. Each time a workspace is imported, My webMethods Server creates a copy of the workspace, with all its content, into each user’s personal folder. A user’s personal folder is located within Folders > Users; you can view it using the system administrator user interface.

➤ To import workspaces

1. In My webMethods: **Navigate > Applications > Administration > System-Wide > Workspace Management.**
2. In the Result panel, click **Import.**
3. In the Import Workspaces window, click **Browse.**
4. In the Choose File to Upload window, navigate to and select the .cdp file that contains the workspaces you want to import and click **Open.**
5. In the Import Workspaces window, click **Apply.**

About the My webMethods Tools Navigation

To add content to a workspace, you add tools to the workspace. The **Tools** tab in the My webMethods navigation lists the tools that you can add. You add a tool by dragging it onto the workspace canvas. For more information about adding content to a workspace, see *Working with My webMethods*.

When you add a tool to a workspace, My webMethods adds a new portlet to the workspace that is specific to the added tool. Out-of-the-box, several tools are provided for end users. For information, see [“Workspace Tools Available by Default” on page 429](#).

System administrators can replace the existing tools taxonomy with a custom taxonomy, which can include additional tools or have some of the standard tools removed. For more information, see [“Customizing the My webMethods Workspace Tools” on page 430](#).

Workspace Tools Available by Default

My webMethods provides tools for the end user to incorporate into a workspace. The tools are available in the **Workspace Tools** section of the **Tools** tab in the My webMethods navigation pane. For a description of the workspace tools and their behavior, see *Working with My webMethods*.

The following table lists the workspace tools that are available by default:

Tool	Usage
Attachments	Enables the user to attach files to a workspace.
Bookmarks	Enables the user to add links to other workspaces and to web sites.
Directory Browser	Enables the user to find identification information about other My webMethods users.
HTML Text	Enables the user to add formatted text to a workspace.
Image	Enables the user to add an image to a workspace.

Tool	Usage
Note	Enables the user to add simple, unformatted text to a workspace.
User Calendar	Enables the user to add a user's calendar to a workspace.

Note:

It is possible to log in as sysadmin and customize the Workspace Tools list. Also, if you are working with an upgraded installation of My webMethods Server, some tools may be present that have been deprecated or removed with newer versions.

Customizing the My webMethods Workspace Tools

System administrators can update the My webMethods tools navigation to modify or completely replace the standard My webMethods tools with a custom taxonomy.

In your customization, you can use existing tools, including the default tools that are listed on the **Tools** tab of the system administrator user interface when in editing a page. Many of the available system administrator page editing tools are described in the *My webMethods Server Portlet Reference*. You can also add custom tools that you create and deploy to My webMethods Server.

Note:

Tools are portlets. You can create portlets using Software AG Designer.

➤ To customize the My webMethods Workspace Tools navigation

1. As system administrator, create a new folder that will be the tools root folder and will hold your custom taxonomy. Add the new folder to this location: Folders > System > Palette Registry.
2. If you want sections in your tools taxonomy, create folders in your custom tools root folder. The name you assign each folder will become a section name on the **Tools** tab of the My webMethods navigation.
3. To add existing tools to your taxonomy, use the Administration Dashboard > Content > Publish portlet to publish instances of folders, forms, links, or portlets into the custom tools taxonomy.

For example, you can publish existing tools that are available to My webMethods Server system administrators when in page editing mode.

4. If you want to include custom tools, create them and save them in a folder within your custom tools root folder.

A tool can be any portlet. You can create custom tools using Software AG Designer and then deploy them to My webMethods Server.

5. Configure the properties of the Leftnav page that My webMethods Server uses for the My webMethods navigation to point it to your new, custom tools root folder.

- a. Navigate to this location: Folders > System > Shell Sections > Noodle Shell Leftnav > Leftnav.
- b. In the row for the Leftnav page, select , and then **Properties**.
- c. In the **Tools Root** field, type the name you assigned your custom tools root folder.

Note:

The name you specify in the **Tools Root** field must be a folder or page that resides in Folder > System > Palette Registry.

- d. Click **Apply**.

24 Customizing Skins

■ What Are Skins?	434
■ Creating and Modifying a New Skin	436
■ Using the Skin Administration Page	439
■ Make-up of a Skin Package	448

What Are Skins?

The look and feel of a page is encapsulated in a skin. You can associate a skin with a particular user or group, in which case they view the contents of a page using that skin. You can associate a skin with a particular folder hierarchy. All users who view pages within that hierarchy view them using the skin.

When you customize a skin, you can modify its look, such as to:

- Brand the page with your own corporate logo
- Change the color scheme to match your corporate colors
- Adopt a look and feel similar to the one used within your company

As a system administrator, you can perform simple customization of skin properties (images, colors, and fonts) using the Skin Administration page of My webMethods Server. To perform more sophisticated customization, you need to directly edit the components that define the skin.

My webMethods Server offers a variety of ways to configure personalization rules that dictate what skin is displayed for a given user, group, or resource. See [“Managing Skin Rules” on page 361](#).

How Skins Use Inheritance

A skin can inherit properties from a parent skin. When you create a customized skin, first make a copy of an existing skin. Through inheritance, you need only identify the ways the custom skin is different from its parent.

Inheritance for skins follows these rules:

- A skin can have only one parent, but the parent skin can also have a parent. In this way, a skin could have a list of ancestors from which it inherits properties.
- A parent skin can have any number of children that inherit properties from it.
- Each skin has a `skin.properties.xml` file that lists the properties the skin does not inherit from its parent. In turn, if this skin has a child skin, the child inherits these modified properties.
- The `skin.properties.xml` file also can contain a “parent” property that identifies the parent of the skin.
- In addition to the properties in the `skin.properties.xml` file, a child also inherits the stylesheets and HTML fragments of the parent skin.
- If a skin does not have a parent identified in the `skin.properties.xml` file, it inherits the properties of the My webMethods Server default skin, which is identified by the `skin.default` alias.

Choosing How Much Customization to Use

When you customize skins, the preferred method is to create a new skin inherited from an existing skin and then modify the new skin. Do not modify the skins that are installed with My webMethods

Server. When you create a new skin and then export it for editing outside the server, you can then deploy it to every server where it is to be used.

You can use one or more of the following techniques to do customization from the simple to the complex:

- **Modify fonts, colors, and images**

The Skin Administration page enables you to create and delete custom skins, and modify the use of images, colors, and fonts. This is the only method of customization you can use without exporting a skin and editing files that make up the skin package. See [“Using the Skin Administration Page” on page 439](#).

Even if you plan to make more extensive modifications, you can create a custom skin, make changes to fonts, colors, or images in the Skin Administration page, and then export the skin for further editing.

- **Use an extended.css file ([“Cascading Style Sheet Definitions” on page 455](#)) to modify skin properties you cannot manage with the Skin Administration page.**

This customization is suitable if you want to make small modifications to existing properties in a skin package. To use an extended.css file, you need to define it in the skin.properties file of the skin.

- **Use a skin.properties.xml file ([“The Skin Properties File” on page 449](#)) if you want to modify values of existing properties but are not adding new properties to the custom skin.**

For example, you can change the value for a font or color already defined in a parent skin. In this case, the skin.properties.xml file overrides properties in the parent skin, but continues to use the CSS of the parent.

- **Use an extended.csi file (dynamic CSS file) ([“Cascading Style Sheet Definitions” on page 455](#)) to add new CSS classes to the custom skin. To use an extended.csi file, you need to define it in the skin.properties file of the skin.**

Using an extended.csi file to create new classes is the simplest way to add new styles to a custom skin.

For definitions of CSS and dynamic CSS files, see [“Cascading Style Sheets” on page 454](#)

To create a new skin from an existing skin, see [“Creating and Modifying a New Skin” on page 436](#).

To export a skin from to your computer for editing, see [“Exporting a Skin to Your Computer” on page 437](#).

How Do I Know What to Modify?

If modifications to a custom skin package are to be more extensive than you can achieve using the Skin Administration page (fonts, colors, and images), you need to know what properties or CSS classes you want to modify from among the hundreds of properties that exist in a skin package. One way to do this is to use a browser developer tool such as the Firebug Add-on for Mozilla Firefox.

Open My webMethods Server in the browser and navigate to a page that uses the skin package you want to examine. Using the developer tool, you can explore the user interface and determine the CSS classes used to display it. With this knowledge, you override values in the parent skin through the use of `extended.css` or `extended.csi` files in the custom skin.

Creating and Modifying a New Skin

As a system administrator, you can create a skin with the Skin Administration page. After you have created the new skin, you can customize it using the Skin Administration page or by manually editing the skin package.

> To create a new skin

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration > Create New Skin.**
2. In the **System Name** field, type a short name that contains only letters, numbers, and the underscore character.

This name is used internally by the server.

3. In the **Display Name** field, type the skin title that you want users to see.

This name has no character restrictions.

4. From the **Parent Skin** list, choose the skin from which the new skin will inherit any unspecified properties.

The system default skin is selected by default.

5. Click **Create**.

A new skin initially inherits all of its properties (colors, fonts, and images) from its parent. You can modify the new skin, changing just a single property, such as adding a new header logo, or you can modify the skin to create a radical new look and feel with completely different colors, fonts, and images.

Having created a new skin, you can begin to customize it by doing one or more of the following:

- Modify the use of images, colors, and fonts, described in [“Using the Skin Administration Page” on page 439](#).

Note:

You should make any planned changes in the Skin Administration page *before* using either of the manual editing techniques.

- Export the skin for manual editing, described in [“Exporting a Skin to Your Computer” on page 437](#).

- Import the skin into Designer for manual editing, described in [“Using Designer to Modify Skin Packages” on page 438](#).

Example:

Assume you want to create a new skin from the parent skin Pearls.

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration > Create New Skin**.
2. In the **System Name** field, type a name such as `wm_skin_carbon`.
3. In the **Display Name** field, type the skin title, `Carbon Sailing Skin`.
4. From the **Parent Skin** list, choose the Pearls skin.
This is the skin from which the new skin will inherit any unspecified properties.
5. Click **Create**.

Exporting a Skin to Your Computer

If you intend to modify a skin package using the Skin Administration page ([“Using the Skin Administration Page” on page 439](#)), you should do so before performing manual editing.

As a system administrator, you can use the Skin Administration page to export a skin package to your computer for manual editing. The skin package is a zip file with a `.skin` file extension.

Tip:

Another way to perform manual editing on a skin package is to import the skin into Designer. See [“Using Designer to Modify Skin Packages” on page 438](#).

» To export a skin

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration > Create New Skin**.
2. Click the Tools icon  for the skin to be exported and then click **Export**.
3. Choose the **Save File** option and click **OK**.
My webMethods Server downloads the file to the default download destination for your browser.
4. Unzip the skin package so you can perform manual editing.

Using Designer to Modify Skin Packages

If you intend to modify a skin package using the Skin Administration page, you should do so before exporting the file and performing manual editing.

One way to modify files in a custom skin package is to import the skin into Software AG Designer and make changes in the UI Development perspective. You can export the skin package directly from My webMethods Server, edit CSS, dynamic CSS, or XML files, and then deploy the skin back to the server without unzipping the skin package.

Note:

This topic is not intended to provide details of working in Software AG Designer. For specific information on working in the MWS Admin and Servers views of Designer, see the Composite Application Framework online help.

➤ To modify custom skin packages in Designer

1. In the MWS Admin view of the UI Development perspective, connect to the instance of My webMethods Server (create a data provider) that contains the custom skin package.

The MWS Admin view contains a tree view of folders and other My webMethods Server assets.

2. Expand the Skins folder, right-click the custom skin package and click **Import/Export > Extract Asset into Project**.
3. In the Extract Asset into Project wizard, click **New**.

4. In the **Project Name** field, type the name of the skin package exactly as it appears in the MWS Admin view and click **Finish** twice.

Designer creates an MWS Skin Project and extracts the assets of the skin package into it.

5. Use either the Navigator view or Project Explorer view to locate CSS, dynamic CSS, or XML files in the skin project, and open the files in a text editor.
6. To deploy the skin package to an instance of My webMethods Server, use the Servers view to connect to the server and then publish the skin package.

To test the skin package, you need to create a skin rule that will trigger the use of the custom skin. See [“Managing Skin Rules” on page 361](#).

Note:

Do not test the skin package using the Preview Server in Designer. That server instance does not have the skin rules needed to properly display the skin.

Using this technique, you can make incremental changes and publish the results to the server periodically for confirmation. Associate the custom skin with a test page so you can perform design work without affecting other users.

Using the Skin Administration Page

The Skin Administration page of My webMethods Server enables you to create and delete custom skins, and modify the use of images, colors, and fonts. To use the Skin Administration page, you must either have the system administrator grant you permission to access the page, or you must log in as the system administrator.

If you want to do more extensive customization than is available through the Skin Administration page, you may still find it convenient to create the custom skin on this page before making modifications.

My webMethods Server applies any changes that you make to images, colors, or fonts to the skin.properties.xml file that is part of the skin package.

Deleting a Skin

As a system administrator, you can delete a skin with the Skin Administration page.

> To delete a skin

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. To delete a skin, click the Tools icon  and then click **Delete**.

Replacing Images in a Skin

The skin of a page contains images, such as logos, that help shape the appearance and structure of the page. Using the Skin Administration page, you can replace images in a skin with images from your local drive, images from an existing skin, or images from a website.

> To replace an image in a skin

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. To modify a skin, click  and then click **Edit**.
3. Click **Images**.
4. Click the skin property for the image you want to replace.

The property is highlighted with a red box.

5. Depending on the source of the image, do one of the following:

- On the local drive - in the **Picker** panel, click **Choose File**. Browse to the new image, click **Open**, and then click **Upload**. To replace the image, click ↶.
- Part of an existing skin - in the **Select palette** list, select the skin from which you want to copy the image. Select the new image and click ↶.
- On a website - in the **Select palette** list, select **URL**, type the website URL and click **OK**. Select the new image and click ↶.

6. Click **Preview**.

The preview demonstrates how your changes have affected the skin.

Tip:

To preview a page other than My Folders, My Home Page, or Public Folders, see [“Previewing a Page Elsewhere on the Server”](#) on page 448.

7. Close the Preview window and click **Save**.

Example:

Assume you have the Carbon Sailing skin installed and want to change from the webMethods logo to the Carbon Sailing logo, carbon-logo.png, which exists on your local file system.

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. Click  for the Carbon Sailing Skin and click **Edit**.
3. Click **Images**.
4. Click the **logo.gif** skin property.

The property is highlighted with a red box.

5. In the **Picker** panel, click **Choose File**, browse to the carbon-logo.png image, click **Open**, and then click **Upload**.

The new image appears in the **Picker** panel.

6. To replace the image, click ↶.
7. In a similar way, select the **banner-bg.gif** property and replace the image with the carbon-bg.jpg file.

The resulting combination results in a complete banner.

8. In the **Preview** panel, choose **My Home Page** from the list and click **Preview**.
9. Close the Preview window and click **Save**.

Replacing Colors Using a Color Picker

There are a number of different color settings in a skin that affect different parts of a page. To change settings, you select colors in the skin editor, apply the colors to skin properties, and preview your changes.

➤ To replace the color of a skin property using a color picker

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. To modify a skin, click the Tools icon  and then click **Edit**.
3. Click **Colors**.
4. Click the skin property for which you want to replace the color.

The skin property is highlighted with a red box.

5. In the **Picker** panel, click the color to be used as a replacement.

The selected color appears in the horizontal bar at the bottom of the **Picker** panel, and the hexadecimal value appears in the **#** field at the top.

Tip:

If you know the hexadecimal value of the replacement color, you can type it directly in the **#** field.

6. To save a color to the **Scratchpad** panel, click the down arrow directly beneath the **Picker** panel.

Tip:

If you want lighter and darker shades of a particular color for use in foregrounds and backgrounds, save the color in the **Scratchpad** panel and, in the vertical bar on the right edge of the **Picker** panel, click above or below the original color. Save multiple colors to the **Scratchpad**.

7. To set the color for a skin property, click  from either the **Picker** panel or the **Scratchpad** panel.
8. At the bottom of the page, click **Preview**.

The preview demonstrates how your changes have affected the skin.

Tip:

If the page is not in one of the folders labeled My Folders, My Home Page, or Public Folders, see [“Previewing a Page Elsewhere on the Server” on page 448](#).

9. Close the Preview window.
10. Click **Save**.

Example:

Assume you have created the Carbon Sailing skin as described in [“Creating and Modifying a New Skin” on page 436](#) and want to change the colors used for buttons.

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. To modify a skin, click  and then click **Edit**.
3. Click **Colors**.
4. Click the **button - button text** skin property.
5. In the **Color** field of the **Picker** panel, replace the default color (BUTTON if the parent skin is Pearls) by typing FFFFFFFF.
6. Click  from the **Picker** panel.

The check box in the **Inherited** column is cleared, indicating this skin property is no longer inherited from the parent skin.

7. Click the **button-bg - button background** skin property.
8. In the **Picker** panel, replace the default color by typing 9EB6C7.
9. Click  from the **Picker** panel.
10. In a similar way, replace the remaining button color properties with the following values:

The following table lists the values to supply for each skin property:

Property	Value
button-border - button border	668899
button-border-light - button border	BBDDEE
button-hover - button text	FFFFFFF
button-hover-bg - button background	4C7499
button-hover-border - button border	395874
button-hover-border-light - button border	ABC9D8
button-disabled - disabled button text	C5C7C7
button-disabled-bg - disabled button background	FFFFFFF
button-disabled-border - disabled button border	C5C7C7

11. At the bottom of the page, click **Preview**.

The following table displays the preview results for the modified button properties:

Default button:	
New color:	
New hover color:	
Final button:	

The final button includes font changes made in [“Replacing Fonts Using a Picker”](#) on page 444.

12. Click **Save**.

Replacing Colors from a Skin or Website

You can replace the color of a skin property using color values from an existing skin or from a website.

➤ To replace the color of a skin property using a color from a skin or a website

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. To modify a skin, click the Tools icon  and then click **Edit**.
3. Click **Colors**.
4. Click the skin property for which you want to replace the color.

The skin property is highlighted with a red box.

5. Depending on the source of the color, do one of the following:
 - Part of an existing skin - In the **Select palette** list, select the skin from which you want to copy the color. Select the new image and click .
 - On a website - In the **Select palette** list, select **URL**, type the website URL and click **OK**. Select the new color and click .
6. At the bottom of the page, under the **Preview** heading, click **Preview**.

The preview demonstrates how your changes have affected the skin.

Tip:

If the page is not in one of the folders labeled My Folders, My Home Page, or Public Folders, see [“Previewing a Page Elsewhere on the Server” on page 448](#).

7. Close the Preview window.
8. Click **Save**.

Replacing Fonts Using a Picker

There are a number of different font settings in a skin that affect different parts of a page. To change settings, you select font settings in the skin editor, apply the settings to skin properties, and preview your changes.

Note:

Font properties used by individual skins can vary. The properties described in the following sample procedure may not be available in all skins.

➤ **To replace the font settings using a picker**

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. To modify a skin, click  and then click **Edit**.
3. Click **Fonts**.
4. Click the skin property for which you want to replace the font settings.

The skin property is highlighted with a red box.

5. In the **Picker** panel, click the color to be used as a replacement.

The following table lists the changes you can make to the font values:

Font settings	Selections
Font-family	Font family. You can select first through tenth choice, enabling the browser to use fonts available on the computer.
Font-size	Select from a list of relative size values or specify a numeric value.
Font-style	Select bold, italic, or underline.
Capitalization	Select from a list of capitalization styles.
Text-spacing	Specify values to be used for letter spacing, word spacing, and line height.

6. To set the font values for a skin property, click  from either the **Picker** panel or the **Scratchpad** panel.
7. At the bottom of the page, click **Preview**.

The preview demonstrates how your changes have affected the skin.

Tip:

If the page is not in one of the folders labeled My Folders, My Home Page, or Public Folders, see [“Previewing a Page Elsewhere on the Server” on page 448](#).

8. Close the Preview window.
9. Click **Save**.

Example:

Assume you have created the Carbon Sailing skin as described in [“Creating and Modifying a New Skin” on page 436](#) and want to change the font values used for buttons.

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. To modify a skin, click  and then click **Edit**.
3. Click **Fonts**.
4. Click the **button - button text** skin property.
5. Under **Font-family**, in the **Primary** list, select the **Tahoma** font.
6. In the **Primary** list, select the **sans-serif**.
7. Under **Font-style**, select **Bold**.
8. Click  from the **Picker** panel.

The check box in the **Inherited** column is cleared, indicating this skin property is no longer inherited from the parent skin.

9. At the bottom of the page, click **Preview**.

The following table displays the preview results for the modified button properties:

Default button:	
New font value:	
Final button:	

The final button includes color changes made in “[Replacing Colors Using a Color Picker](#)” on [page 441](#).

10. Click **Save**.

Replacing Fonts from a Web Site

In controlling the font families used by a skin, you have the choice of designing the style yourself or selecting a set of font families used by another Web site. The following procedure presents a scenario describing how you might go about capturing and using families from another site.

Note:

Font properties used by individual skins can vary. The properties described in the following sample procedure may not be available in all skins.

➤ **To replace the font families used by a skin with the font families used by another Web site**

1. As system administrator: **Administration Dashboard > User Interface > Skin Administration**.
2. To modify a skin, click the  and then click **Edit**.
3. Click **Fonts**.
4. On the **Select palette** list, select **URL**, type the Web site URL and click **OK**.

The Palette displays a list of font families derived from the Web site.

5. In the Palette on the lower-right side of the page, find a line that seems to represent the normal body text of the Web site, and select it.

This action should result in several font families being listed in the **Picker** panel. If not, try some other lines in the palette that look like normal body text.

6. Click  directly above the Palette.

This action saves the font information to the **Scratchpad** panel for later use.

7. In the line you just created in **Scratchpad** panel, select the text in the edit field and rename it to something meaningful, such as **my regular text**.
8. In the **Skin Properties** list on the left side of the page, select the **regular** skin property.

The **regular** skin property is highlighted with a red box, meaning it is selected for editing.

9. To set the **regular** skin property, click  from the **Scratchpad** panel.

This action sets the **regular** skin property with the value selected in the **Scratchpad** panel.

10. In the **Skin Properties** list on the left side of the page, select the **bold** skin property.

The **bold** skin property is highlighted with a red box.

11. In the **Scratchpad** panel, select **my regular text**.

Because the **my regular text** scratchpad item was already the active item in the **Scratchpad** panel (with a dark red border around it), selecting it again makes it the active item for the page (with a bright red border around it), and loads the **my regular text** font information back into the **Picker** panel.

12. In the **Font-style** area of the **Picker** panel, select the **Bold** check box.

13. Click  from the **Picker** panel.

This action sets the **bold** skin property with the value in the **Picker** panel.

14. In the **Skin Properties** list, click the **small** skin property.

15. In the **Scratchpad** panel, select **my regular text**.

16. In the **Font-size** area of the **Picker** panel, do one of the following:

- Select **Relative size**, and then change the value in the list to be a size smaller than the existing size
- Select **Numeric size**, and then edit the number in the field to be a value two or three smaller than the existing size

17. Click the arrow directly to the left of the **Picker** panel.

18. In the **Skin Properties** list, select the **medium** skin property.

19. In the **Scratchpad** panel, select **my regular text**.

20. In the **Preview** list at the bottom of the page, choose **Public folders** and then click **Preview**.

The preview demonstrates how your changes have affected the skin.

Tip:

If the page is not in one of the folders labeled My Folders, My Home Page, or Public Folders, see [“Previewing a Page Elsewhere on the Server” on page 448](#).

21. Close the preview window.

22. Click **Save**.

Previewing a Page Elsewhere on the Server

You can preview a page other than My Folders, My Home Page, or Public Folders in the skin editor.

➤ To preview a page elsewhere on the server

1. Make changes to skin properties within Skin Administration.
2. In the **Preview** list at the bottom of the page, choose **URL**.
3. Open a new browser window, navigate to the server, navigate to the page you want to preview.
4. In the **Address** bar of the browser, select the URL and type CTRL+C to copy it.
5. Return to the Script Prompt dialog from the first browser and paste in the URL by typing CTRL+V.
6. Click **OK**.
7. Click **Preview**.

Make-up of a Skin Package

There are two ways to make a skin package available for editing.

- [“Exporting a Skin to Your Computer” on page 437](#)
- [“Using Designer to Modify Skin Packages” on page 438](#)

With a skin package exposed for editing, you can see the files and directories that make up the package:

- The `skin.properties.xml` file describes the components implemented by the skin. You can modify fonts, colors, and images using the Skin Administration page, but for other skin components, you need to edit the skin properties file directly. For more information, see [“The Skin Properties File” on page 449](#).

If you have created a new skin in My webMethods Server but not yet made any changes, the only property in the file may be the parent skin from which all properties are inherited.

- The `skinDeploy.xml` file contains deployment information for the skin package. For more information, see [“The Skin Deployment File” on page 453](#).
- The `css` directory. Cascading Style Sheets (CSS) describe the properties that make up a skin. A generated CSS for a skin is derived from the `skin.properties.xml` file as modified by other style sheet files. For more information, see [“Cascading Style Sheets” on page 454](#).

If you have created a new skin in My webMethods Server but not yet made any changes, there may be no Cascading Style Sheets in the directory because all properties are inherited from the parent skin. If you add or modify stylesheets, you should place them in this directory.

- The components directory. This directory contains proprietary files you are not likely to modify as part of a skin package.
- The images directory. If you add or modify images used in the skin, you should place them in this directory.

The Skin Properties File

The `skin.properties.xml` file describes the components implemented by the skin. Any changes you make to a skin using the Skin Administration page ([“Using the Skin Administration Page” on page 439](#)) are reflected in the `skin.properties.xml` file for the skin. The `base.csi` file for a skin references the properties in the `skin.properties.xml` file for use when My webMethods Server generates the `base.css` file.

The Importance of the Skin Properties File

The `base.csi` file for a skin references properties defined in the `skin.properties.xml` file. My webMethods Server uses the `base.csi` file to generate the `base.css` file that governs how the skin is displayed. This is a powerful mechanism that enables you to create new skins based on a common ancestor by specifying images, fonts, and colors in a new `skin.properties.xml` file without having to re-create a new stylesheet for each skin.

In the following examples, the Pearls skin is the a parent. No CSS (`.css` or `.csi`) change is required in the custom skin.

Replacing images

You can change the banner background image in a custom skin by modifying the value of the `images/banner-bg.gif` property in the `skin.properties.xml` file for the custom skin.

The following table lists the properties that you modify when replacing images:

File	Contents
parent skin.properties.xml	<pre><property> <name>images/banner-bg.gif</name> <value>images/dot.gif</value> <description>banner background</description> </property></pre>
parent base.csi	<pre>background-image: url(@skin images/banner-bg.gif);</pre>
child skin.properties.xml	<pre><property> <name>images/banner-bg.gif</name> <value>images/carbon-bg.jpg</value> </property></pre>

Also, see [“How do I replace one image with another?”](#) on page 452.

Globally replacing fonts

You can globally change the fonts used in a custom skin by modifying the value of the `fonts/regular` property in the `skin.properties.xml` file for the custom skin.

The following table lists the properties that you modify when replacing fonts:

File	Contents
parent skin.properties.xml	<pre><property> <name>fonts/regular</name> <value>font-family: tahoma, sans-serif; font-size: 0.7em;</value> <description>standard text</description> </property></pre>
parent base.csi	<pre>@skin fonts/regular;</pre>
child skin.properties.xml	<pre><property> <name>fonts/regular</name> <value>font-family: Georgia, 'Times New Roman', Times, serif; font-size: 1em; font-size: 0.7em; </value> <description>standard text</description> </property></pre>

Globally replacing colors

You can globally change the colors used in a custom skin. In this example, the `colors/section-body-border` property controls the color of row borders. One change in the `skin.properties.xml` file for the custom skin, affects seven properties in the `base.csi` file.

The following table lists the properties that you modify when replacing colors:

File	Contents
parent skin.properties.xml	<pre><property> <name>colors/section-body-border</name> <value>#ccc</value> <description>row border</description> </property></pre>
parent base.csi	<pre>.tbl { border: @skin sizes/section-body-border; solid @skin colors/section-body-border;; . . }</pre>

File	Contents
child skin.properties.xml	<pre><property> <name>colors/section-body-border</name> <value>#dfe6ec</value> <description>row border</description> </property></pre>

Also, see [“How do I modify colors?”](#) on page 452.

Making Entries in a Skin Properties File

If you edit the `skins.properties.xml` file manually, you do not need to include components inherited from one of the skin’s ancestors. Rather, you need only include modifications to inherited components. A good way to see the properties that describe a skin is to export the parent skin package using the method described in [“Exporting a Skin to Your Computer”](#) on page 437, unzip the skin package, and examine the `skin.properties.xml` file.

Note:

Modifying the skin properties in the `skins.properties.xml` file requires CSS expertise.

The `skin.properties.xml` file is made up of property elements, each describing a component of the skin. The property elements have this format:

```
<property>
  <name>property_name</name>
  <value>property_value</value>
  <description>optional_description</description>
</property>
```

The following guidelines apply to the property element:

- If the value is the same as the name, you can omit the value.
- The description is optional.
- You can add comments using standard XML comment syntax:

```
<!-- This is a comment -->
```

A skin component can have multiple properties associated with it. For example, the banner that appears at the top of a My webMethods Server page has over twenty properties that determine its appearance, such as:

Color, position, and padding of the background

Color, font, and weight of link text, selected links, including hover characteristics

Images and their positioning

To see the properties that describe a skin, export the pearls skin package using the method described in [“Exporting a Skin to Your Computer”](#) on page 437, unzip the skin package, and examine the `skin.properties.xml` file. The following examples describe a few ways to modify properties.

How do I specify a parent skin?

To make the development of a custom skin easier, you need to specify a parent skin from which the custom skin inherits its properties. A skin can have only one parent skin. In the following example, the `skin.properties.xml` file specifies the pearls skin package as the parent of the custom skin:

```
<!-- parent skin; all unspecified properties are inherited -->
<property>
  <name>parent</name>
  <value>skin.wm_skin_pearls</value>
  <description>parent skin</description>
</property>
```

How do I replace one image with another?

To replace one image with another, you first need to move a copy of the new image into the `images` directory for the skin package. Then you need to locate the property for the image in the `skins.properties.xml` file.

For example, perhaps you want to rebrand the page by changing the logo image that appears in the left side of the banner. The new image has the name `my_logo.png`.

```
<!-- images -->
<property>
  <name>images/logo.gif</name>
  <value>images/my_logo.png</value>
  <description>header logo</description>
</property>
```

How do I modify colors?

The topic [“Replacing Colors Using a Color Picker”](#) on page 441 shows several colors modified in the Skin Administration page. The same changes look like this in the `skin.properties.xml` file.

```
<property>
  <name>colors/button</name>
  <value>#FFFFFF</value>
  <description>button text</description>
</property>
<property>
  <name>colors/button-bg</name>
  <value>#9EB6C7</value>
  <description>button background</description>
</property>
<property>
  <name>colors/button-border</name>
  <value>#668899</value>
  <description>button border</description>
</property>
<property>
  <name>colors/button-border-light</name>
  <value>#BBDDEE</value>
  <description>button border</description>
```

```

</property>
<property>
  <name>colors/button-hover</name>
  <value>#FFFFFF</value>
  <description>button text</description>
</property>
<property>
  <name>colors/button-hover-bg</name>
  <value>#4C7499</value>
  <description>button background</description>
</property>
<property>
  <name>colors/button-hover-border</name>
  <value>#395874</value>
  <description>button border</description>
</property>
<property>
  <name>colors/button-hover-border-light</name>
  <value>#ABC9D8</value>
  <description>button border</description>
</property>
<property>
  <name>colors/button-disabled</name>
  <value>#C5C7C7</value>
  <description>disabled button text</description>
</property>
<property>
  <name>colors/button-disabled-bg</name>
  <value>#FFFFFF</value>
  <description>disabled button background</description>
</property>
<property>
  <name>colors/button-disabled-border</name>
  <value>#C5C7C7</value>
  <description>disabled button border</description>
</property>

```

How do I add Cascading Style Sheets?

You can add multiple style sheets to a skin.properties.xml file. For more information, see [“Adding Stylesheets to a Skin Package” on page 457](#).

The Skin Deployment File

The skinDeploy.xml file contains deployment information for the skin package. This deployment information is contained entirely as attributes of the root wm_xt_skin element.

Note:

It is unnecessary to make any modifications to this file for a newly created custom skin package.

The following table lists the attributes of the wm_xt_skin element in the skinDeploy.xml file

Attribute	Description
name	The name of the skin package.

Attribute	Description
alias	A comma-separated list of aliases to the skin as used by My webMethods Server.
description	The display name for the skin as it appears in the user interface.
version	The version number of the skin. When you modify and release the skin package, you should increment the minor version number.
cssPreview	A semicolon-separated list of CSS style declarations that demonstrates the style on an html element in the user interface.
serversideResourcePath	A legacy attribute. The value of this attribute should always take the form <code>/ui/skins/skin-name/</code> where <code>skin-name</code> is the skin's system name.
clientsideResourcePath	A legacy attribute. The value of this attribute should always take the form <code>/ui/skins/skin-name/</code> where <code>skin-name</code> is the skin's system name.

Example

The Carbon Sailing skin package used as an example elsewhere in this guide has a `skinDeploy.xml` file that looks like this:

```
<wm_xt_skin name="wm_skin_carbon"
  alias="skin.wm_skin_carbon,skin.carbon"
  description="Carbon Sailing Skin"
  version="1.0"
  cssPreview="color:#fff; background-color:#6f6f60;"
  serversideResourcePath="/ui/skins/wm_skin_carbon/"
  clientsideResourcePath="/ui/skins/wm_skin_carbon/">
</wm_xt_skin>
```

Cascading Style Sheets

Note:

Modifying the files described in this topic requires CSS expertise.

When My webMethods Server displays a page, the skin for that page uses a Cascading Style Sheet (CSS) that describes all of the properties making up the skin. My webMethods Server generates this base CSS at the time you deploy the skin package to the server and every time the server starts. The generated CSS file is based on properties that can be provided from a number of sources:

- Properties inherited from a hierarchy of parent skins
- A `skins.properties.xml` file that is part of the skin package
- One or more CSS or dynamic CSS files in the skin package

My webMethods Server uses dynamic CSS files to reference property values in the `skin.properties.xml` file and place them in the generated `base.css` file for a skin package. See [“How Dynamic CSS Files Work” on page 456](#).

Cascading Style Sheet Definitions

base.css

The primary CSS for a skin package. A `base.css` file most often exists only as a generated CSS file, having been defined by other sources in the skin package or its parent. The `base.css` can be modified by one or more extended CSS or dynamic CSS files. It is not recommended that you create a `base.css` file for a custom skin package, but if you do so, define it in the `skin.properties.xml` file. You cannot define both a `base.css` file and a `base.csi` file in the same custom skin package.

extended.css

A secondary CSS for a skin package. Entries in the extended CSS file override the `base.css` file. You can have multiple extended CSS files, such as to provide browser-specific CSS values. Extended CSS files, each with a different name, are defined in the `skin.properties.xml` file.

base.csi

A dynamic CSS file that references properties in the `skin.properties.xml` file ([“How Dynamic CSS Files Work” on page 456](#)). As a result of the `base.csi` file, the generated `base.css` file contains properties from the `skin.properties.xml` file. The `skin.properties.xml` file for a top-level skin package (one that has no parent) describes all properties that make up a skin. If there is no `base.csi` file in a skin package, it is inherited from a parent skin. The `base.csi` file is defined in the `skin.properties.xml` file. You cannot define both a `base.css` file and a `base.csi` file in the same custom skin package.

extended.csi

A secondary dynamic CSS file for a skin package ([“How Dynamic CSS Files Work” on page 456](#)). Entries in the extended `csi` file override the `base.csi` file. You can have multiple extended `csi` files, such as to provide browser-specific CSS values. The extended `csi` files, each with a different name, are defined in the `skin.properties.xml` file.

How Do I Choose Which Type of Stylesheet to Modify?

Use these guidelines in determining which stylesheet to use:

- Use an `extended.css` file to make selected modifications to a custom skin that is not significantly different from the parent skin on which it is based. An `extended.css` file overrides:
 - The `base.css` or `base.csi` file in the parent skin package
- If you define an `extended.csi` file for a custom skin, the classes in it override only the corresponding classes in the `base.css` or `base.csi` file of the parent skin.
- Use an `extended.csi` file to change selected references to a `skin.properties.xml` file, either in the custom skin package or in its parent. An `extended.csi` file overrides:
 - The `base.csi` file in the parent skin package
 - The `base.csi` file in the custom skin package, if you have created one.

Using an `extended.csi` file to create new classes is the simplest way to add new styles to a custom skin.

As an alternative to defining `extended.css` files in the `skin.properties.xml` file, you can use `import` statements to include them in an `extended.csi` file. For example, the following statements cause two CSS files to be included during generation of the `base.css` for a skin:

```
@import url(@skin css/general.css);◆"
@import url(@skin css/dialog_styles.css);◆
```

- Use a `base.csi` file if you have made extensive modifications to the `skin.properties` file in the custom skin package that require a revised set of references. A `base.csi` file is required in any skin package that does not have a parent skin package.

If you define a `base.csi` file for a custom skin, it completely overrides any `base.css` or `base.csi` file in the parent skin. If the custom `base.csi` file contains a subset of the classes defined for the parent skin, the generated `base.css` file will contain only that same subset of classes.

How Dynamic CSS Files Work

My webMethods Server replaces skin property references in a dynamic CSS file with property values in the `skin.properties.xml` file and places them in the generated `base.css` file for a skin package. A top-level skin package (one that has no parent) typically has no `base.css` file, instead using the `base.csi` file to create the generated `base.css` file.

Used in custom skin packages, this feature makes it possible to create new skins based on a common ancestor by simply specifying skin properties in a new `skin.properties.xml` file. If you make simple modifications to the `skin.properties.xml` file that do not require changes to the dynamic CSS file, you can use the `base.csi` file inherited from the parent skin. If you make more extensive modifications, you may have to create an `extended.csi` file to properly reference the changed or added properties.

Note:

Modifying the files described in this topic requires CSS expertise.

Dynamic CSS files make use of at-rules (or `@rules`) to reference properties in the local `skin.properties.xml` file or properties inherited from the parent skin. For example, if your `base.csi` file contains this rule:

```
body {
    @skin fonts/regular;
    color: @skin colors/text;;
    background-color: @skin colors/text-bg;;
    direction: @skin dir/text;;}
```

and the `skin.properties.xml` files contains these property definitions:

```
<property>
  <name>fonts/regular</name>
  <value>font-family: trebuchet; font-size: 20px;</value>
</property>
<property>
  <name>colors/text</name>
  <value>#000</value>
```

```

</property>
<property>
  <name>colors/text-bg</name>
  <value>#fff</value>
</property>
<property>
  <name>dir/text</name>
  <value>ltr</value>
</property>

```

the generated `base.css` file will contain this rule:

```

body {
  font-family: trebuchet; font-size: 20px;
  color: #000;
  background-color: #fff;
  direction: ltr;
}

```

If you create a `base.csi` or `extended.csi` file for use with a skin package, you need to declare it in the `skin.properties.xml` file, as described in [“Adding Stylesheets to a Skin Package” on page 457](#).

Adding Stylesheets to a Skin Package

You can use CSS and dynamic CSS files to modify the generated `base.css` file for a skin package. To add a CSS or dynamic CSS file to a skin package, you define it in the `skin.properties.xml` file for the custom skin package. Each stylesheet is defined in a property element having this form:

```

<property>
  <name>css/name.css</name>
  <value>css/name.extension</value>
  <description>optional description</description>
</property>

```

If the file name in the `<value>` element is the same as the `<name>` element, you can omit the `<value>` element.

If the `skin.properties.xml` file does not have property element for a stylesheet, it inherits the stylesheet in the parent skin.

Note:

When you add or modify a stylesheet, you should place it in the `css` directory of the skin package.

Example: base.css

Note:

We do not recommend that you create a `base.css` file for use in a custom skin package. However, you can use the syntax here to compare with the examples that follow.

You can have only one `base.css` element in the `skin.properties.xml` file. In the following example, you could omit the `<value>` element because it is identical to the `<name>` element.

```

<property>
  <name>css/base.css</name>
  <value>css/base.css</value>

```

```
</property>
```

Example: base.csi

If you want to make extensive changes that override the base.csi file of a parent skin, you might create a new base.csi file and define it in the custom skin package. In this case, the <name> element is base.css and the <value> element is base.csi, as shown here:

```
<property>
  <name>css/base.css</name>
  <value>css/base.csi</value>
</property>
```

Example: extended.css or extended.csi

The rules for extended.css and extended.csi files are the same as those for base.css:

```
<property>
  <name>css/extended.css</name>
  <value>css/extended.css</value>
</property>
```

or:

```
<property>
  <name>css/extended.css</name>
  <value>css/extended.csi</value>
</property>
```

Example: Multiple extended.css or extended.csi Files

A skin.properties.file can have properties for multiple extended.css or extended.csi files at the same time. One use for this capability is to provide browser-specific properties.

```
<property>
  <name>css/extended.css</name>
  <value>css/extended.csi</value>
</property>

<property>
  <name>css/ie6.css</name>
  <value>css/ie6.csi</value>
  <description>special ie6 stylesheet rules</description>
</property>

<property>
  <name>css/ie7.css</name>
  <value>css/ie7.csi</value>
  <description>special ie7 stylesheet rules</description>
</property>
```

25 Working with Shells in My webMethods Server

■ What Are Shells?	460
■ Creating a New Shell	460
■ Modifying a Shell	461
■ Inserting Extra Tags into the HTML <head> Element	462
■ Using an Alias with a Shell Section	462
■ Deleting a Shell	463
■ Making an Empty Shell Section	463

What Are Shells?

My webMethods Server derives the content and layout of the header and footer of a page from the content and layout of the current shell's Header and Footer folders. When the server renders one page, it is actually displaying the contents of up to five folders at once: the shell's Header, the shell's Leftnav, the requested page, the shell's Rightnav, and the shell's Footer. The server displays the content of the requested page as individual portlets, but it renders the content of shell sections without title bars, borders, or additional spacing. You can apply Cascading Style Sheet (CSS) classes and styles to the rows and columns that compose the shell sections, as well as specify the exact dimensions of those rows and columns, to further customize the layout of the shell.

The Titlebar shell section applies a title bar to each portlet on the page, which includes the display name and buttons for controlling the portlet. You can hide the Titlebar shell section individually for each portlet.

You can create and modify a shell with the Shell Administration page. To use the Shell Administration page, you must either have the system administrator grant you permission to access the portlet, or you must log in as the system administrator.

Creating a New Shell

The first step in constructing a new shell is to create it using an existing shell as a parent. A new shell initially inherits all of its properties from its parent. These properties (or shell sections) are: Header, Footer, Leftnav, Rightnav, and Titlebar. You can replace any of these sections with a new, custom shell section.

> To create a new shell

1. As system administrator: **Administration Dashboard > User Interface > Shell Administration > Create New Shell.**

2. In the **Name** field, type a name for the shell.

This name has no character restrictions.

3. (Optional) In the **Description** field, type a description of the shell.

The description appears in the list of shells on the Shell Administration page.

4. From the **Parent Shell** list, choose the shell from which the new shell will inherit any unspecified properties.

The system default shell is selected by default.

5. Click **Create**.

Modifying a Shell

After you have created a new shell from a parent shell, you can modify individual sections of that shell independently to construct a new shell.

> To modify a shell

1. As system administrator: **Administration Dashboard > User Interface > Shell Administration.**
2. To modify a shell, click  and then click **Properties.**
3. If you want to change the display name for the shell, in the **Display Name** field, type a new name for the shell.
4. If you want to change the parent shell from which to take the various shell sections, in the **Parent** list, choose the parent shell. The list contains shells that currently exist on the server.
5. For each shell section, specify one of the following parents or sources:
 - **Inherited** - use the shell section from the shell chosen in the **Parent Shell** field.
 - **Portal Page** - use the content of an existing folder for the shell section, move the folder to the **Selected Items** box and then click **Select.** Not available for the **Titlebar** shell section.
 - **Portlet** - To use an existing portlet, move the portlet to the **Selected Items** box and then click **Select.** Available only for the **Titlebar** shell section.
6. For the shell sections that you want to edit, do one of the following:
 - To edit a shell section inherited from another shell, select **Inherited** and click **Clone from Parent.** My webMethods Server creates a folder based on the inherited shell section.
 - To edit an existing folder used as a shell section, select the **Portal Page** option and make sure the name of the target folder is displayed.

Note:

Within the server, you cannot edit portlets. You need to use Software AG Designer.

7. Click **Edit.**

The folder that represents the shell section opens in edit mode.

8. Modify the shell section just as you would any other folder.
9. Click **Save.**

Inserting Extra Tags into the HTML <head> Element

You can insert JavaScript libraries and stylesheets into the <head> element of a shell. The valid HTML tags are <link>, <meta>, <script>, and <style>.

> To insert into the HTML <head> element

1. As system administrator: **Administration Dashboard > User Interface > Shell Administration**.
2. To modify a shell, click  and then click **Properties**.
3. In the **Extra Tags for HTML <head>** field, type the HTML tags that add custom code to the shell. For example:

```
<link href="default.css" rel="stylesheet" type="text/css"
```

4. Click **Save**.

Using an Alias with a Shell Section

If a folder has an alias, you can use the alias to select it for use as a shell section. For information about using aliases with folders, see [“Managing Aliases” on page 324](#).

> To select a shell section using an alias

1. As system administrator: **Administration Dashboard > User Interface > Shell Administration**.
2. To modify a shell, click  and then click **Edit**.
3. In the shell section you want to associate with an alias, click **Use Alias**.
4. In the **Alias Name** field of the server resource selector, type the alias of the folder you want to use for this shell section.
5. To determine if the server can find the alias, click **Test**.
6. If the server correctly resolves the alias, click **Select**.
7. If needed, you can clone this folder or edit it directly, as described in [“Modifying a Shell” on page 461](#).
8. Click **Save**.

Deleting a Shell

After it is no longer needed, you can delete a shell.

> To delete a shell

1. As system administrator: **Administration Dashboard > User Interface > Shell Administration**.
2. To delete a shell, click , and then click **Delete**.

Making an Empty Shell Section

A shell always has the four folders that make up the shell sections. You may, however, want a shell design in which one or more of the shell sections is empty and takes up no space in the display. A shell section is empty if it contains no portlets and has no formatting information associated with it. In the default shells provided with My webMethods Server, the Leftnav and Rightnav shell sections display as being empty.

You cannot edit the Titlebar shell section as you do the others. To hide the title bar, you need to set the **Titlebar** attribute for individual portlets to **No**.

> To make a shell section (other than a Titlebar) empty

1. As system administrator: **Administration Dashboard > User Interface > Shell Administration**.
2. To modify a shell, click , and then click **Edit**.
3. In the **Alias Name** field of the resource selector, type the alias `shell.section.blank`.
4. To determine if the server can find the alias, click **Test**.
5. If the server correctly resolves the alias, click **Select**.
6. Click **Save**.

V Using Command Central to Manage My webMethods Server

26	Administering My webMethods Server	467
27	Using the Command Line to Manage My webMethods Server	483
28	Authenticating My webMethods Server	497

26 Administering My webMethods Server

- Administering My webMethods Server Instances 468
- Configuring My webMethods Server Ports 471
- Configuring Directory Services 472
- Configuring My webMethods Server Email 479
- Working with My webMethods Server Environment Variables 480
- Monitoring KPIs of My webMethods Server Instances 481
- Using Trusted Authentication to Connect to My webMethods Server 482

Administering My webMethods Server Instances

Creating My webMethods Server Instances

➤ To create a My webMethods Server instance using the Command Central web user interface

1. On the Command Central home screen, select an environment and click the **Installations** tab.
2. Click the name of an installation, and then go the **Instances** tab for the installation.
3. On the **Instances** tab, click **Create instance** and select My webMethods Server.

Command Central displays the **Create instance** wizard.

4. On the Specify Properties screen of the Create instance wizard, specify the following:
 - **Instance Name** - this field is required. Specify a name for the new instance that is unique among server instances on the machine.
 - **Register Windows service/UNIX daemon for automatic startup** - when checked, My webMethods Server will be installed as service on Windows machines, and as daemon on UNIX machines. This checkbox is cleared by default.
5. On the **Database** tab, specify the following:

The following table lists the database configurations that you specify when creating a new My webMethods Server instance:

Field	Description
Database type	The type of the My webMethods Server database. The default value is Oracle.
JDBC URL	<p>The database connection URL, as follows:</p> <pre>jdbc:wm:database://hostname:port; databaseName=db_name; option=value;option=value</pre> <p>where <i>database</i> is the type of the My webMethods Server database, <i>hostname</i> is the address of the server where the database is installed, <i>port</i> is the connection port of the database, and <i>db_name</i> is the name of the database that My webMethods Server uses. Specify additional options in <i>option=value</i> pairs, separated by semi-colons. Valid database types are: oracle, db2, sqlserver, mysql, postgresql.</p>

Field	Description
	Example: jdbc:wm:sqlserver://localhost:1433;databaseName=WMDB
Database user	The name of the user to connect to the database.
Password	The password for connecting to the database.
JDBC driver class	The driver class name of the JDBC driver. Required when using a JDBC driver different than the one supplied by Software AG.

6. On the **Ports** tab, specify the following:

The following table lists the port configurations that you specify when creating a new My webMethods Server instance:

Field	Description
HTTP port	Optional. The HTTP port on which the new server instance listens. The default port number is 8585.
HTTPS port	Optional. The HTTPS port on which the new server instance listens. The default port number is 8586.
Debug port	Optional. The Java debug port for the new server instance. The default port number is 10033.
JMX port	Optional. The JMX port for the new server instance. The default port number is 5002.

7. When creating a new instance as a node in a My webMethods Server cluster, on the **Cluster** tab of the **Create instance** wizard, specify the following:
- **Cluster node name** - the name of the cluster node. Specify a name unique for the cluster. Do not include spaces in the name.
 - **JNDI provider URL** - the URL of the Universal Messaging server to use as a JMS provider.
8. Click **Next**, review the summary of the configuration and then click **Finish** to create the new My webMethods Server instance.

Updating My webMethods Server Instances

You update the server instance after you install new components, or modify the configuration of the instance, for example, when adding the instance to a My webMethods Server cluster.

➤ **To update a My webMethods Server instance using the Command Central web user interface**

1. On the **Instances** tab for your installation, select the instance to update by clicking on the instance icon.
2. From the drop-down menu of the gear button, select **Update instance**.
3. Update the instance configuration as required, and click **Next**. You can configure the following:

The following table lists the configurations that you specify when updating a new My webMethods Server instance:

Field	Description
Node name	Optional. The name of the server node that hosts the instance in a My webMethods Server cluster. Use this field to rename an existing server node. Do not include spaces in the node name.
JDBC URL	Optional. Connection URL for the My webMethods Server database.
Database user	Optional. The name of the user to connect to the My webMethods Server database.
Password	Optional. The password for connecting to the My webMethods Server database.

4. Review the instance configuration and click **Finish**.

Deleting My webMethods Server Instances

➤ **To delete a My webMethods Server instance using the Command Central web user interface**

1. On the **Instances** tab for your installation, select the instance to delete by clicking on the instance icon.
2. Click **Remove** to delete the instance.
3. Click **OK** to confirm the deletion.

Pausing and Resuming the Operation of a My webMethods Server Instance

You can pause and resume the operation of My webMethods Server instances using the Command Central web user interface, and command-line interface. When you pause a My webMethods Server instance from Command Central, the instance enters maintenance mode. When an instance is in maintenance mode, the My webMethods login page displays a notification, and only Administrator or sysadmin users can log on.

For more information about pausing and resuming My webMethods Server instances using the command-line interface, see [“Lifecycle Actions for My webMethods Server-ENGINE” on page 488](#).

➤ **To pause or resume a My webMethods Server instance using the Command Central web user interface**

1. On the **Instances** tab of your installation, click the status icon for the instance to pause or resume.
2. From the **Lifecycle Actions** drop down menu, select the required action.
3. Click **OK** to confirm the action.

Configuring My webMethods Server Ports

Configuring My webMethods Server Ports in Command Central

My webMethods Server listens for client requests on one or more ports. When a port receives a message or request, My webMethods Server invokes the appropriate services. Each port is configured to work with a specific protocol. You can associate HTTP or HTTPS with one or more additional ports as needed. By default, My webMethods Server is pre-configured with HTTP at 8585.

The MWS_default component is the OSGi profile. The My webMethods Server component is the standard profile for the server instance. You can edit configuration settings for the My webMethods Server component, but you cannot add or delete them.

Perform the following procedure to configure a My webMethods Server port, using the Command Central web user interface.

➤ **To configure My webMethods Server ports**

1. In the Environments pane, select the environment in which you want to view the My webMethods Server instance.
2. Select the **Instances** tab.
3. Expand the MWS_ *instanceName* node containing the My webMethods Server instance you want to configure.
4. Click **My webMethods Server** in the name column.
5. Select the **Configuration** tab. Make sure **My webMethods Server** is selected in the left pane.
6. Select **Ports** from the drop-down list box.

7. **Test** and **Save** the port.

Editing Port Settings

Perform the following procedure to change the server port settings, using the Command Central web user interface.

> To enable or disable a port

1. Select the My webMethods Server environment from the Environments pane, then click the My webMethods Server instance you want to edit from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select **Ports** in the drop-down list.
4. Click the number of the port you want to edit and click **Edit**. The port settings are now editable. Make the necessary changes to the port settings.
5. **Test** and **Save** the changes.

Configuring Directory Services

My webMethods Server includes an internal directory with default system users. If you are using an external directory (for example LDAP), you can configure My webMethods Server to also access user and group information from that directory through an external directory service. You can create a directory service and connect My webMethods Server to an external LDAP, ADSI, or ADAM sever using the Command Central web user interface.

> To configure an external directory service

1. In the **Environments** pane, select the environment of the My webMethods Server instance you want to configure.
2. Select the **Instances** tab.
3. Expand the `MWS_instanceName` node of the My webMethods Server instance.
4. Click **My webMethods Server**.
5. Select the **Configuration** tab. Make sure **My webMethods Server** is selected in the left pane.
6. Select **LDAP** from the drop-down list.

7. Click , and select the type of directory service to add from the following options:

- **LDAP** - Lightweight Directory Access Protocol.
- **ADSI** - Active Directory Service Interfaces.
- **ADAM** - Active Directory Application Mode.

8. Fill in the form to configure the directory service.

For more information about available settings, see [“LDAP, ADSI, and ADAM Directory Service Properties” on page 473](#).

9. **Test** and **Save** the directory service configuration.

LDAP, ADSI, and ADAM Directory Service Properties

When configuring an external directory service of type LDAP, ADAM, or ADSI, you can configure the following properties.

In the **General** section:

The following table lists the directory service properties you configure in the General section:

Property	Description
Name	Required. The name to identify the external directory service. My webMethods Server uses this name to display the external directory service in the user interface.
Description	A descriptive comment about the external directory service.
Keywords	One or more keywords to use when searching for external directory services.

In the **Cache** section:

The following table lists the directory service properties you configure in the Cache section:

Property	Description
Cache Capacity	Required. The number of database queries to cache. The default is 1000. My webMethods Server deletes the cache entries when the number of cached queries reaches the specified capacity, starting from the oldest entries.

Property	Description
Cache Timeout	<p>Required. The period of time for which queries remain in the cache unless the cache capacity is exceeded. The default is 1 hour.</p> <p>My webMethods Server deletes cache entries when the cache timeout expires, even if the specified cache capacity is not reached.</p>

My webMethods Server saves all cache in memory and clears all cache entries when restarted.

In the **Connection Information** section:

The following table lists the directory service properties you configure in the Connection Information section:

Property	Description
Service Enabled	Enables or disables the directory service. The default is Yes . This service is enabled.
Connection Error Threshold	Required. The maximum number of connection errors to occur before disabling the service. The default is 10.
Auto Reconnect	Attempt to reconnect to the directory server if the service is disabled after reaching the connection error threshold or if the connection to the server is lost due to a network outage or planned maintenance. Enabled by default.
Auto Reconnect Interval	The period of time (in seconds) to wait between subsequent attempts to reconnect. The default is 6.
Provider URL	<p>Required. The URL for the external directory server using the following syntax:</p> <pre>ldap://host_name:port_number</pre>
Base DN	Required. The root distinguished name to use when querying the directory server. For example, <code>ou=mywebMethods,o=webmethods.com</code>
User DN	The additional user DN to use when searching and loading users.
Groups DN	The additional group DN to use when searching and loading users.
Use Kerberos	Whether to use Kerberos authentication when connecting to the LDAP service. The default is No . Do not use Kerberos. For more information about using directory services with

Property	Description
Use Ticket Cache	<p data-bbox="678 260 1474 331">Kerberos, see “Configure Kerberos Authentication for Directory Services” on page 270.</p> <p data-bbox="678 352 1474 598">Whether to use Kerberos credentials cache while the user session lasts. Available only when the LDAP service is configured to use Kerberos Authentication. The default is No. Do not use ticket cache. For more information about configuring Kerberos ticket cache for directory services, see “Configure Kerberos Authentication for Directory Services” on page 270.</p>
Security Principal	<p data-bbox="678 625 1474 724">Required when not using Kerberos Ticket Cache. The distinguished name required to log in to the external directory server.</p>
Security Credentials	<p data-bbox="678 751 1474 829">Required when not using Kerberos Ticket Cache. The password required to log in to the external directory server.</p>
Failover URLs	<p data-bbox="678 856 1474 955">The URL to another LDAP server that My webMethods Server uses for failover if the primary LDAP server, specified in the Provider URL field, fails. Separate multiple values with spaces.</p>
Search Timeout	<p data-bbox="678 982 1474 1081">Required. The maximum amount of time (in seconds) that an LDAP search query can run before it expires. The default is 0 - the query does not expire.</p> <p data-bbox="678 1108 1474 1312">Unless you configure the connection timeout in the custom_wrapper.conf file, My webMethods Server uses the Search Timeout to define the timeout of a connection to an LDAP server. For more information about configuring an LDAP server connection timeout, see “Configuring a Connection Timeout for an LDAP Directory Service” on page 118.</p>
Enable Default Wildcard Searches	<p data-bbox="678 1339 1474 1438">Required. Enables or disables the use of wildcard characters in directory searches. The default is Yes. Enable default wildcard searches.</p> <p data-bbox="678 1470 1474 1585">Disabling wildcard searches might improve performance for large servers. When using wildcards, servers do not use any internal indexes for search performance.</p>
Enable Group Across Directory Service	<p data-bbox="678 1612 1474 1808">Required. Indicates whether to query for group membership across all external directory services, configured in My webMethods Server. When you enable this option, the search queries for group membership across all directory services, which degrades the login performance. The default is No. Group Across Directory Service.</p>

Property	Description
	For more information, see “Group Membership Across Directory Services” on page 119.
Enable GroupQuickSearch	Required for Active Directory. Indicates whether to determine the group membership of an Active Directory user with one query instead of a recursive search. When you enable this option, the search uses one query, which improves the login performance.. Users must belong either to an Active Directory security group, or a regular group. The default is Disabled .
ActiveDirectory Domain URLs	Applies only to Active Directory. Specify multiple Active Directory sub-domain URLs, separated by spaces.

In the **Advanced Object Filters** section:

The following table lists the directory service properties you configure in the Advanced Object Filters section:

Property	Description
User Object Filter	<p>The LDAP filter that My webMethods Server applies to all queries when searching for users. Use a technical LDAP query that limits the type of objects, exposed in My webMethods Server.</p> <p>Note: It is recommended that you examine the My webMethods Server directory debug logs to ensure that the query is working correctly.</p>
Group Object Filter	<p>The LDAP filter that My webMethods Server applies to all queries when searching for groups. Use a technical LDAP query that limits the type of objects, exposed in My webMethods Server.</p> <p>Note: Examine the My webMethods Server directory debug logs to ensure that the query is working correctly.</p>
Use Nested Groups	Enables or disables searches in nested LDAP groups. The default value is No. Do not use nested groups.
Use the Virtual List View Control	Enables or disables the use of the Virtual List View control to retrieve a subset of objects for an LDAP query. The default value is No. Do not use the VLV control. Applies only when the automatic configuration of LDAP server controls is disabled.

Property	Description
Use Server Side Paging Control	Enables or disables the use of the Server-Side Paging control to page the results of an LDAP query. The default value is No. Do not use the Paging control . Applies only when the automatic configuration of LDAP server controls is disabled.
Use Server Side Sorting Control	Enables or disables the use of the Server-Side Sorting control to sort the results of an LDAP query in a particular order. The default value is No. Do not use the Sorting control . Applies only when the automatic configuration of LDAP server controls is disabled.
Automatically Configure Server Side Controls	Enables or disables the automatic configuration of LDAP server controls by My webMethods Server. The default value is Yes. Autoconfigure the controls .

In the **User Attributes** section:

The following table lists the directory service properties you configure in the User Attributes section:

Property	Description
User Object Class	Required. The User Object Class attribute for the external directory service. The default is <code>person</code> .
User ID	Required. The User ID attribute for the external directory service. The default is <code>uid</code> .
First Name	Required. The First Name attribute for the external directory service. The default is <code>sn</code> .
Last Name	Required. The Last Name attribute for the external directory service. The default is <code>givenName</code> .
Full Name	Required. The Full Name attribute for the external directory service. The default is <code>cn</code> .
E-mail Address	Required. The Email Address attribute for the external directory service. The default is <code>mail</code> .
Password	Required. The Password attribute for the external directory service.
User Disabled	The name of an attribute in the external directory service that identifies a user as disabled. The default is <code>true</code> .
User Disabled Value Regex	The regular expression to use when evaluating the User Disabled attribute for the external directory service.

Property	Description
UUID	<p>The name of the attribute to use as a universally unique identification attribute of a user. Specify a string of maximum 128 characters, for example <code>cn</code> or <code>email</code>.</p> <p>Note:</p> <p>If you change the value of UUID for an existing directory service, you must run the <code>UserDirectory_UpdateUUID</code> utility to update the UUID value of directory service users.</p> <p>For more information, see “Configuring Universally Unique Identifier (UUID) for Users” on page 120.</p>

In the **Group Attributes** section:

The following table lists the directory service properties you configure in the Group Attributes section:

Property	Description
Group Object Class	Required. The Group Object Class attribute for the external directory service. The default is <code>groupofuniqueNames</code> .
Group ID	Required. The Group ID attribute for the external directory service. The default is <code>cn</code> .
Group Name	Required. The Group Name attribute for the external directory service. The default is <code>cn</code> .
Group Members	Required. The Group Members attribute for the external directory service. The default is <code>uniqueMember</code> .
Group E-mail	Required. The Group Email attribute for the external directory service. The default is <code>mail</code> .

In the **Connection Pool** section:

The following table lists the directory service properties you configure in the Connection Pool section:

Property	Description
Minimum Connections	The minimum number of connections to the external directory server to keep open at all times. The default is 1.
Maximum Connections	The maximum number of connections to the external directory server to keep open at all times. The default is 20.

Property	Description
Maximum Connection Time	The maximum amount of time to keep a connection to the external directory server open, before recycling the connection. The server resets this value for each LDAP search to ensure that an LDAP connection remains open during the search process. The default is 10 minutes.
Clean Up Interval	The time interval for cleaning up expired LDAP connections. The default is 1 minute.

Note:

In some LDAP implementations, the paging cookie is bound to a specific LDAP connection. Make sure that the value for the Maximum Connections property is large enough to handle concurrent LDAP searches and the value for the Maximum Connection Time property is long enough to ensure that searches can finish within the specified time range.

Configuring My webMethods Server Email

Perform the following procedure to configure My webMethods Server email, using the Command Central web user interface.

➤ To configure email

1. Select the My webMethods Server environment from the Environment pane, then click the instance from the **Instances** tab.
2. Click the **Configuration** tab.
3. Select **Email** in the drop-down list.

Command Central displays the My webMethods Server SMTP Server Configuration.

4. Click **Edit**.
5. In **Connection Basics**, specify:

The following table lists the basic connection configurations:

Field	Description
Server Name	The SMTP server's host name. For example: smtp.server.com.
Port	The SMTP server's port number.
Sender Name	The default name to use in the From field of the email messages sent by the server.

Field	Description
Sender Email	The default email address to use in the From field of the email messages sent by the server.

6. Expand **Advanced Settings** and specify:

The following table lists the advanced email configurations:

Field	Description
SMTP Username	Optional. The user name that My webMethods Server has to supply for authentication. If the SMTP server requires authentication, specify the user name.
SMTP Password	Optional. The password associated with the SMTP Username . If the SMTP server requires authentication, specify the appropriate password.

7. Click **Test** and **Save** the email settings.

Working with My webMethods Server Environment Variables

Considerations when Configuring My webMethods Server Variables in Command Central

You should consider the following naming conventions when configuring My webMethods Server environment variables:

- The length of the variable name and value must not exceed 255 characters.
- Variable names are case-sensitive.
- No restrictions for special characters apply to the variable name.

Configuring My webMethods Server Variables

When adding a My webMethods Server environment variable in the Command Central user interface, select one of the following Custom types:

- **Default.** Creates variables without secure fields.
- **Secure.** Creates password variables whose values are encrypted by My webMethods Server.

The following table lists the fields to complete when adding or editing My webMethods Server global environment variables in the Command Central user interface:

Field	Description
Key	<p>Required. The name of the My webMethods Server global environment variable. My webMethods Server uses the key to refer to the environment variable while performing environment variable substitution.</p> <p>Note: The Key field is disabled when editing a variable.</p>
Type	<p>Required. The data type of the My webMethods Server global environment variable.</p> <p>Note: The Type field is disabled when editing a variable.</p>
Value	<p>The value of the My webMethods Server global environment variable.</p> <p>When you add or edit a secure environment variable, the value you specify in this field is hidden. After you save the environment variable, the Value field remains empty.</p>

Important:

After you edit or delete an environment variable, the webMethods applications using the variable are automatically restarted.

Monitoring KPIs of My webMethods Server Instances

➤ To view the KPIs of My webMethods Server instances in the Command Central web user interface

1. On the Environments pane, select the environment you want to monitor.
2. Click the **Instances** tab.
3. Select the My webMethods Server you want to monitor.
4. Click the **Overview** tab.

The **Monitoring** section in the **Dashboard** shows the KPIs of the My webMethods Server instance.

The following table lists the KPIs that My webMethods Server returns:

Name	Marginal Value	Critical Value	Maximum Value
Number of user sessions	80% of maximum	95% of maximum	At least 100, or high water mark. (High water mark is the highest value ever reached.)
JDBC connection pool size (maximum number of connections to JDBC)	80% of maximum	95% of maximum	As configured.
Average response time (in milliseconds)	50% of maximum	90% of maximum	At least 10 seconds, or high water mark.

Using Trusted Authentication to Connect to My webMethods Server

By default Command Central version 9.8 and higher uses SAML-based trusted authentication when connecting to My webMethods Server. You do not need to perform any operation to enable this functionality.

However, if you change the value of the **Assertion Parameter Name** field in the My webMethods Server administration web user interface, you must configure an additional Java parameter, `MWS.SAML.paramname`, in the `custom_wrapper.conf` file of the Platform Manager that manages the My webMethods Server run-time component. The `MWS.SAML.paramname` parameter specifies the changed Assertion Parameter Name value. For more information about setting the Assertion Parameter Name value, see *Administering My webMethods Server*.

➤ To configure the `MWS.SAML.paramname` parameter in Platform Manager

1. Go to the `Software AG_directory \profiles\SPM\configuration` directory and open the `custom_wrapper.conf` file in a text editor.
2. Add the following parameter:

```
wrapper.java.additional.number=
-DMWS.SAML.paramname=assertionParameterName
```

where *number* is a unique sequential number depending on the already existing parameter numbers in the file, and *assertionParameterName* is the modified value of the **Assertion Parameter Name** field in the My webMethods Server administration user interface.

3. Save the file and restart Platform Manager.

27 Using the Command Line to Manage My webMethods Server

■	Commands that My webMethods Server Supports	484
■	Configuration Types that My webMethods Server-ENGINE Supports	486
■	Lifecycle Actions for My webMethods Server-ENGINE	488
■	My webMethods Server Instance Management	489
■	Run-time Monitoring Statuses for My webMethods Server-ENGINE	493
■	Run-time Monitoring States for My webMethods Server	494

Commands that My webMethods Server Supports

The following table lists all Platform Manager commands that My webMethods Server supports, and where to find additional information about each command.

Commands	Additional Information
sagcc create configuration data	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using this command, see “Configuration Types that My webMethods Server-ENGINE Supports” on page 486.</p>
sagcc delete configuration data	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc get configuration data	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using this command, see “Configuration Types that My webMethods Server-ENGINE Supports” on page 486.</p>
sagcc update configuration data	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc get configuration instances	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc list configuration instances	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using this command, see “Configuration Types that My webMethods Server-ENGINE Supports” on page 486.</p>
sagcc get configuration types	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using this command, see “Configuration Types that My webMethods Server-ENGINE Supports” on page 486.</p>
sagcc list configuration types	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using this command, see “Configuration Types that My webMethods Server-ENGINE Supports” on page 486.</p>

Commands	Additional Information
sagcc exec configuration validation create	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using this command, see “Configuration Types that My webMethods Server-ENGINE Supports” on page 486.</p>
sagcc exec configuration validation delete	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc exec configuration validation update	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using this command, see “Configuration Types that My webMethods Server-ENGINE Supports” on page 486.</p>
sagcc create instances	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using the command, see “Creating a My webMethods Server Instance” on page 489.</p>
sagcc delete instances	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc list instances supported products	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc update instances	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using the command, see “Updating a My webMethods Server Instance” on page 492.</p>
sagcc get inventory components	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc list inventory components	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc update inventory components	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>
sagcc exec lifecycle	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p>

Commands	Additional Information
	For My webMethods Server-specific information about using this command, see “Lifecycle Actions for My webMethods Server-ENGINE” on page 488.
sagcc get monitoring	<p>For general information about the command, see <i>Software AG Command Central Help</i>.</p> <p>For My webMethods Server-specific information about using this command, see:</p> <ul style="list-style-type: none"> ■ “Run-time Monitoring Statuses for My webMethods Server-ENGINE” on page 493 ■ “Run-time Monitoring States for My webMethods Server” on page 494

Configuration Types that My webMethods Server-ENGINE Supports

The My webMethods Server-ENGINE run-time component supports creating instances of the configuration types listed in the following table:

Configuration Type	Use to configure...
COMMON-ADMINUI	<p>Full URL to My webMethods Server.</p> <ul style="list-style-type: none"> ■ If the My webMethods Server HTTP port is enabled, use the following format where <i>hostname</i> is the My webMethods Server host name and <i>httpport</i> is the My webMethods Server HTTP port number. <p><code>http://hostname:httpport</code></p> <ul style="list-style-type: none"> ■ If the My webMethods Server HTTPS port is enabled and the HTTP port is <i>not</i> enabled, use the following format where <i>hostname</i> is the My webMethods Server host name and <i>httpsport</i> is the My webMethods Server HTTPS port number. <p><code>https://hostname:httpsport</code></p>
COMMON-CLUSTER	<p>Settings for a My webMethods Server cluster. You can configure the load balancer, JMS provider, and node roles for a My webMethods Server cluster. For information about the fields and the values to specify, see <i>Administering My webMethods Server</i>.</p>

Configuration Type	Use to configure...
COMMON-JDBC	<p>Important: Any changes to the cluster configuration take effect only after you restart the node.</p> <p>The default connection pool for the My webMethods Server database connection. You can use the command line interface to edit the pool, but not delete it.</p> <p>You can also add, update, or delete additional custom JDBC pools that custom Composite Application Framework (CAF) applications running on My webMethods Server use.</p> <p>Note: You can manage instances of this configuration type using the command line interface, but not the Command Central user interface.</p>
COMMON-KEYSTORES	<p>Keystores for My webMethods Server. You can edit the keystores that My webMethods Server uses for its HTTPS port to provide your own keystores.</p> <p>Note: You can manage instances of this configuration type using the command line interface, but not the Command Central user interface.</p>
COMMON-LOCAL-USERS	<p>Settings for managing internal users of a My webMethods Server instance. COMMON-LOCAL-USERS-<i>userId</i> supports configuring the details for each user. For information about the fields and values to specify, see <i>Administering My webMethods Server</i>.</p>
COMMON-PORTS	<p>The My webMethods Server HTTP and HTTPS ports.</p> <p>When adding, editing, and removing ports, keep the following in mind:</p> <ul style="list-style-type: none"> ■ Ensure at least an HTTP or an HTTPS port is defined. ■ You can only delete the HTTPS port if the HTTP port is defined. If you delete the HTTPS port, you can later add it again. ■ You can only delete the HTTP port if the HTTPS port is defined. If you delete the HTTP port, you can later add it again. ■ There are no restrictions with regards to editing port numbers.

Configuration Type	Use to configure...
	<p>Note: You can also manage ports in the Command Central user interface if the ports are enabled. After enabling the ports in My webMethods Server Cluster Administration and restarting My webMethods Server, the ports are visible in the Command Central user interface.</p>
COMMON-SMTP	Settings for sending e-mail messages.
COMMON-TRUSTSTORES	Truststores for My webMethods Server. You can edit the truststores that My webMethods Server uses for its HTTPS port to provide your own truststores.
	<p>Note: You can manage instances of this configuration type using the command line interface, but not the Command Central user interface.</p>
COMMON-VARS	Global environment variables for My webMethods Server.

Lifecycle Actions for My webMethods Server-ENGINE

You can perform lifecycle operations for My webMethods Server using the Software AG Command Central command-line utility, or web user interface. For more information about using the web user interface, see [“Pausing and Resuming the Operation of a My webMethods Server Instance” on page 470](#).

For general information about the `sagcc exec lifecycle` command, see *Software AG Command Central Help*.

The following table lists the actions My webMethods Server-ENGINE supports with the `sagcc exec lifecycle` command and the status that the My webMethods Server run-time component returns when an action is executed:

Action	Description
pause	Puts the run-time component in maintenance mode. When successful, the run-time status is set to NOT_READY.
resume	Takes the run-time component out of maintenance mode. When successful, the run-time status is set to ONLINE.

For more information about the run-time statuses that the My webMethods Server-ENGINE run-time component reports, see [“Run-time Monitoring Statuses for My webMethods Server-ENGINE” on page 493](#).

My webMethods Server Instance Management

Creating a My webMethods Server Instance

The following table lists the parameters to include when creating a My webMethods Server instance using the Command Central instance management commands:

Command	Parameter	Description
sagcc create instances	MwsProgramFiles	Required. The product ID of the installed My webMethods Server.
	instance.name= <i>name</i>	Required. The name of the new My webMethods Server instance. The default name is <code>default</code> .
	fixRepository= <i>repoID</i>	Optional, but recommended. Specifies the ID of the fix repository, registered in Command Central, which you used to install fixes during the initial product installation. If you use this parameter, Command Central re-installs the fixes from this repository required for the new product instance you are creating. Note that if you do not include this parameter, Command Central does not apply fixes on the new product instance.
	db.type= <i>type</i>	Required. The type of database used by the new server instance. Specify one of the following databases: <ul style="list-style-type: none"> ■ sqlserver or ms - Microsoft SQL Server ■ oracle - Oracle ■ db2 - DB2 ■ mysql_ee - MySQL Enterprise Edition ■ mysql_ce - MySQL Community Edition ■ postgresql - PostgreSQL
	db.url=" <i>url</i> "	Required. The connection URL for the database. Enclose the <i>url</i> value in double quotes.
	db.username= <i>name</i>	Required. The user name assigned to the My webMethods Server database.

Command	Parameter	Description
	<code>db.password=password</code>	Required. The password of the My webMethods Server database user.
	<code>[install.service={true false}]</code>	Optional. Specifies whether to install a new My webMethods Server instance as an application or a service. Valid values are: <ul style="list-style-type: none"> ■ <code>true</code> - install as a service. ■ <code>false</code> - install as an application (default).
	<code>db.driver=name</code>	Optional. The class name of the JDBC driver used to connect to the My webMethods Server database.
	<code>node.name=name</code>	Optional. The name of the cluster node that will host the new server instance. The default name is <code>master</code> .
	<code>http.port=port</code>	Optional. The port number on which the new server instance listens. The default port number is 8585.
	<code>https.port=port</code>	Optional. The https port on which the new server instance listens. A value of 0 disables the listener.
	<code>debug.port=port</code>	Optional. The Java debug port for the new server instance. The default port number is 10033.
	<code>jmx.port=port</code>	Optional. The JMX port for the new server instance. The default port number is 5002.
	<code>server.features={core default fabric all}</code>	Optional. The set of component features configured with the new server instance. Specify one of the following values: <ul style="list-style-type: none"> ■ <code>core</code> - The minimum set of features needed to support development of JSR 168 portlets using Software AG Designer. This value indicates a pure runtime, with a single skin and shell, and no administration or configuration features. ■ <code>default</code> - The standard set of My webMethods Server features, but without the additional features or development tools found in the <code>Software AG_directory \MWS\ components</code> directory. This is the default value. ■ <code>fabric</code> - The default My webMethods Server taxonomy with all installed My webMethods Server user interfaces.

Command	Parameter	Description
		<ul style="list-style-type: none"> ■ <code>all</code> - The standard set of features plus all components found in the <i>Software AG_directory \MWS\components</i> directory.
	<code>components.include=name</code>	Optional. Components in the <i>Software AG_directory \MWS\components</i> directory to include in the new server instance.
	<code>components.exclude=name</code>	Optional. Components in the <i>Software AG_directory \MWS\components</i> directory to exclude from the new server instance.
	<code>components.override={true false}</code>	<p>Optional. Specifies whether to overwrite the component files in the <i>Software AG_directory \MWS\server\serverName\deploy</i> directory that are older than the component files in the <i>Software AG_directory \MWS\components</i> directory. Valid values are:</p> <ul style="list-style-type: none"> ■ <code>true</code> - overwrite component files in the <code>\deploy</code> directory. ■ <code>false</code> - do not overwrite component files in the <code>\deploy</code> directory. This is the default value.

Examples When Executing on Command Central

- To create a new instance for an installed My webMethods Server with instance name “test1”, database type “sqlserver”, database user name “mws”, database password “mws”, and database URL “jdbc:wm:sqlserver://localhost:1433;databaseName=testDB” in the installation with alias name “local”:

```
sagcc create instances local MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
```

- To create a new instance for an installed My webMethods Server with instance name “test1”, database type “sqlserver”, database user name “mws”, database password “mws”, database URL “jdbc:wm:sqlserver://localhost:1433;databaseName=testDB”, database driver “com.wm.dd.jdbc.sqlserver.SQLServerDriver”, http port “9595”, and JMX port “7000” in the installation with alias name “local”:

```
sagcc create instances local MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
db.driver=com.wm.dd.jdbc.sqlserver.SQLServerDriver http.port=9595
jmx.port=7000
```

- To create a new instance for an installed My webMethods Server with instance name “test1”, http port “9595”, and JMX port “7000” in the local installation and register the instance as a

service. The new My webMethods Server instance uses an external database with database type “sqlserver”, database user name “mws”, database password “mws”, database URL “jdbc:wm:sqlserver://localhost:1433;databaseName=testDB” and database driver “com.wm.dd.jdbc.sqlserver.SQLServerDriver”:

```
sagcc create instances local MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
db.driver=com.wm.dd.jdbc.sqlserver.SQLServerDriver http.port=9595
jmx.port=7000 install.service=true
```

Examples When Executing on Platform Manager

- To create a new instance for an installed My webMethods Server with instance name “test1”, database type “sqlserver”, database user name “mws”, database password “mws”, and database URL “jdbc:wm:sqlserver://localhost:1433;databaseName=testDB”:

```
sagcc create instances MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
```

- To create a new instance for an installed My webMethods Server with instance name “test1”, database type “sqlserver”, database user name “mws”, database password “mws”, database URL “jdbc:wm:sqlserver://localhost:1433;databaseName=testDB”, database driver “com.wm.dd.jdbc.sqlserver.SQLServerDriver”, http port “9595”, and JMX port “7000”:

```
sagcc create instances MwsProgramFiles instance.name=test1
db.type=sqlserver db.username=mws db.password=mws
db.url="jdbc:wm:sqlserver://localhost:1433;databaseName=testDB"
db.driver=com.wm.dd.jdbc.sqlserver.SQLServerDriver http.port=9595
jmx.port=7000
```

Updating a My webMethods Server Instance

The following table lists the parameters to include when updating a My webMethods Server instance using the `sagcc update instances` command:

Parameter	Description
MwsProgramFiles-	Required. The ID of the My webMethods Server instance that you want to update. For example: MwsProgramFiles-test1.
components.override={true false}	Optional. Specifies whether to overwrite the component files in the <i>Software AG_directory</i> \MWS\server\serverName\deploy directory that are older than the components files in the <i>Software AG_directory</i> \MWS\components directory. Valid values are: <ul style="list-style-type: none"> ■ true - overwrite component files in the \deploy directory.

Parameter	Description
	<ul style="list-style-type: none"> ■ <code>false</code> - do not overwrite component files in the <code>\deploy</code> directory. This is the default value.
<code>dont.update.classpath={true false}</code>	<p>Optional. Specifies whether to update the generated My webMethods Server classpath. Valid values are:</p> <ul style="list-style-type: none"> ■ <code>true</code> - do not update the server classpath. ■ <code>false</code> - update the server classpath. This is the default value.
<code>platformdir=path</code>	<p>Optional. The path to the platform installation directory. The default path is <i>Software AG_directory</i>.</p>
<code>node.name=name</code>	<p>Optional. The name of the cluster node that hosts the My webMethods Server instance. Specify a new value to rename the cluster node when updating the instance.</p>
<code>timeout=number</code>	<p>Optional. The time interval in seconds to wait for the command to complete before terminating the attempt to execute the command.</p>

For more information about using the `sagcc update instances` command, see *Software AG Command Central Help*.

Deleting a My webMethods Server Instance

You can delete a My webMethods Server instance using the `sagcc delete instances` command. For more information about using the `sagcc delete instances` command, see *Software AG Command Central Help*.

Run-time Monitoring Statuses for My webMethods Server-ENGINE

The following table lists the run-time statuses that the My webMethods Server-ENGINE run-time component can return in response to the `sagcc get monitoring state` command, and the meaning of each run-time status:

Run-time Status	Meaning
ONLINE	<p>The My webMethods Server-ENGINE run-time component is running.</p> <p>The run-time component indicates ONLINE when the profile JVM is running and that the My webMethods Server port is responding.</p>

Run-time Status	Meaning
ERROR	The My webMethods Server-ENGINE run-time component is not running due to some failure, and attempts to start it again have failed. The run-time component indicates ERROR when My webMethods Server cannot connect to the database, or fails to install a component.
NOT_READY	The My webMethods Server-ENGINE run-time component is not ready. The run-time component indicates NOT_READY when the server is in maintenance mode.
STARTING	The My webMethods Server-ENGINE run-time component is starting. The run-time component indicates STARTING when the server starts for the first time, or is starting and reports an HTTP 503 status.
STOPPED	The My webMethods Server-ENGINE run-time component is stopped.
UNKNOWN	The status of The My webMethods Server-ENGINE run-time component cannot be determined.

Run-time Monitoring States for My webMethods Server

In response to the `sagcc get monitoring state` command, My webMethods Server provides information about key performance indicators (KPIs). The following table lists the KPIs, valid for My webMethods Server:

KPI	Description
Number of user sessions	Use this KPI to monitor the number of active user sessions so that you can take corrective actions if the number approaches a critical value. <ul style="list-style-type: none"> ■ Marginal is 80 active user sessions. ■ Critical is 95 active user sessions. ■ Maximum is 100 or more active user sessions.
Number of active connections in the JDBC pool	Use this KPI to monitor the number of active connections in the JDBC pool so that you can take corrective actions if the number of connections approaches a critical value. <ul style="list-style-type: none"> ■ Marginal is 90 active connections. ■ Critical is 95 active connections. ■ Maximum is 100 or more active connections.

KPI	Description
Average response times in milliseconds	<p data-bbox="613 260 1479 359">Use this KPI to monitor My webMethods Server response times so that you can take corrective actions if the response times slow to a critical value.</p> <ul data-bbox="613 386 1479 533" style="list-style-type: none"><li data-bbox="613 386 1479 422">■ Marginal is 5000 milliseconds.<li data-bbox="613 443 1479 478">■ Critical is 9000 milliseconds.<li data-bbox="613 499 1479 533">■ Maximum is response times at 10000 or more millisecond.

28 Authenticating My webMethods Server

- Changing the Authentication Mode for My webMethods Server 498
- Enabling Access for Administrative User Accounts for My webMethods Server in Command Central 498
- Using Unix Shell Scripts to Change Connection Credentials for Managed Products 499
- Verifying the Outbound Authentication Settings 499
- Accessing Administrative Interfaces Through Command Central 499

Changing the Authentication Mode for My webMethods Server

By default, in Command Central the Authentication mode for run-time components that support trusted authentication is set to **Trusted**.

On the instance **Overview** tab, click  in the **Authentication** field to change the authentication mode using the Authentication Mode dialog box.

When you specify the authentication mode for an instance, that authentication mode is also set for all layered product instances of the main product instance. However, changing the authentication mode for the OSGI profile of My webMethods Server does not change the authentication mode for the My webMethods Server run-time component that belongs to that OSGI profile.

To use basic authentication, you must change the authentication mode for a run-time component to **Fixed User**. Command Central uses basic authentication with a fixed user to communicate with Platform Manager. With **Fixed User** authentication, the authentication credentials for the Platform Manager will be fixed.

Enabling Access for Administrative User Accounts for My webMethods Server in Command Central

To install managed products Software AG Command Central requires that you change the password for the ADMINISTRATOR credentials alias for My webMethods Server that corresponds to the default administrator user SysAdmin. All other administrative users are disabled by default. After installation you can provide strong passwords and enable the administrative accounts, such as Administrator, Designer, and WEBM_SYSUSER in the Command Central web user interface. After changing the Administrator password for My webMethods Server in Command Central, the outbound credentials are updated automatically.

➤ To change passwords and enable system users for My webMethods Server in Command Central

1. In the Environments pane in Command Central, select the environment that contains the managed product instance.
2. In the Instances table, select the My webMethods Server component under *MWS-instanceName*, for example MWS-default.
3. On the Configuration tab, select **Internal Users**.
4. On the Users page, click the user name for the account you want to enable.
5. Click **Edit** and specify a new password.

6. For **Enabled**, select **Yes**.

Using Unix Shell Scripts to Change Connection Credentials for Managed Products

You can use the following sample Unix shell script to configure basic authentication credentials for product components managed by Command Central.

```

NODE_ALIAS=local
USERNAME=Administrator
PASSWORD=secret
RCID=integrationServer-default
# RCID=MwsProgramFiles-default
# RCID=Universal-Messaging-nirvana
# RCID=OSGI-CTP
# RCID=OSGI-InfraDC

sagcc get configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS-Administrator

-o administrator.xml
sed "s,/>,><Password>${PASSWORD}</Password></User>,g" administrator.xml >
  administrator_new.xml
sagcc update configuration data $NODE_ALIAS $RCID COMMON-LOCAL-USERS
-Administrator -o administrator_new.xml

# verify connection
sagcc get monitoring runtimestatus $NODE_ALIAS $RCID -e ONLINE

```

Verifying the Outbound Authentication Settings

Use the following steps to verify that Command Central is configured with the correct outbound authentication settings.

➤ To verify that Command Central is configured with the correct user credentials

1. In Command Central, on the **Overview** tab for the product component, click . Check that the product status is **Online** and the JVM KPIs are updated.
2. On the **Logs** tab, check the product log for authentication errors.

Accessing Administrative Interfaces Through Command Central

In Command Central, single sign-on (SSO) is designed to manage webMethods products using an administrative link without any post-installation configuration. When performing advanced

configuration tasks, you might need to access the product's primary administrative interface. Command Central provides a link to the administrative interface on the Instances Overview page for each managed product. For example, when you click the My webMethods link on the Overview page of a My webMethods Server instance, Command Central redirects the browser to the corresponding My webMethods URL.

Important:

Use *only* one Command Central instance to manage a landscape. You cannot access the Command Central web user interface for a Command Central instance from another Command Central instance.

For information about generating and configuring custom SSO and SAML certificates for Software AG Common Platform-based products, see the Software AG Security Infrastructure documentation.