## Oracle® Fusion Middleware

Configuration Guide for IBM WebSphere Application Server

11*g* Release 1 (11.1.1)

**E17764-01**

January 2011

This document describes how to use the Oracle Fusion Middleware Configuration Wizard or Oracle-specific wsadmin commands to create or modify IBM WebSphere cells for Oracle Fusion Middleware products.

ORACLE®

Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server, 11*g* Release 1 (11.1.1)

E17764-01

# Contents

## 4  Using wsadmin to Configure Oracle Fusion Middleware

## 5  wsadmin Command Reference

# List of Examples

## List of Figures

## List of Tables

x

# Preface

This document describes how to use the Oracle Fusion Middleware Configuration Wizard to configure Oracle Fusion Middleware products in IBM WebSphere cells on a UNIX or Windows operating system.

## Audience

This document is intended for system administrators and application developers who are configuring one or more installed Oracle Fusion Middleware products in an IBM WebSphere environment.

## Document Organization

This document is organized as follows:

- Chapter 1, "Configuration Overview," provides an introduction to IBM WebSphere cells and cell topology, as well as an overview of the Oracle Fusion Middleware Configuration Wizard. It also describes the general workflow for creating cells and federating remote machines to an existing cell. This chapter also describes what happens when you clone an existing server or add a new server to an existing cluster.

- Chapter 2, "Using the Configuration Wizard," describes how to start the Configuration Wizard. This chapter includes screen workflows for creating a new cell, federating a remote node to a cell, or modifying an existing cell. This chapter also includes an example configuration workflow for creating a clustered server environment on multiple nodes.

- Chapter 3, "Configuration Wizard Screens," provides screen shots and detailed descriptions for each of the Oracle Fusion Middleware Configuration screens.

- Chapter 4, "Using wsadmin to Configure Oracle Fusion Middleware," describes how to set up the environment on a machine prior to running `wsadmin` commands. This chapter also provides examples of how to perform various cell configuration tasks using `wsadmin` commands or scripts.

- Chapter 5, "wsadmin Command Reference," describes the `OracleMWConfig` and `OracleMWConfigUtilities` commands for `wsadmin` that are used to configure Oracle Fusion Middleware products in an IBM WebSphere environment.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our

documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

**Accessibility of Code Examples in Documentation**

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

**Accessibility of Links to External Web Sites in Documentation**

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

**Access to Oracle Support**

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

# Related Documents

For more information, see the following documents in the Oracle Fusion Middleware 11*g* Release 11.1.1.3 documentation set:

- *Oracle Fusion Middleware Third-Party Application Server Guide*
- *Oracle Fusion Middleware Domain Template Reference*

# Conventions

The following text conventions are used in this document:

| Convention | Meaning |
| --- | --- |
| **boldface** | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary. |
| *italic* | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values. |
| monospace | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

# 1

# Configuration Overview

This chapter provides a basic overview of the Oracle Fusion Middleware Configuration Wizard for IBM WebSphere and the IBM WebSphere cell environment. It contains the following sections:

- Introduction to IBM WebSphere Cells
- Overview of the Oracle Fusion Middleware Configuration Wizard
- Cell Creation Process
- Federation Process
- Creating Clusters and Cloning Application Servers

## 1.1 Introduction to IBM WebSphere Cells

An IBM WebSphere Application Server (IBM WebSphere AS) cell contains one Application Server. An IBM WebSphere Application Server Network Deployment (IBM WebSphere ND) cell contains one or more Application Servers that can be clustered for high availability and scaled to meet performance needs.

> **Note:** In this document, IBM WebSphere is used to reference both IBM WebSphere AS and IBM WebSphere ND. The specific product names are used when appropriate.

Table 1–1 briefly describes the components of a typical IBM WebSphere cell. Figure 1–1 illustrates the basic topology of an IBM WebSphere ND cell. For more detailed information about IBM WebSphere cells, see the IBM WebSphere documentation.

*Table 1–1    Components of a Cell*

| Component | Description |
| --- | --- |
| Cell Manager node | This is the node from which the Deployment Manager manages all nodes in the cell. A cell has only one Cell Manager node. |
| Deployment Manager | The Deployment Manager is a service that manages all nodes in the cell. |
| Application nodes | A cell must have at least one node on which applications run. This node can be on the same machine as the Deployment Manager, or it can be on a remote machine. |
| | To scale your application environment, add additional remote nodes to the cell by federating machines to the cell. |

*Table 1–1   (Cont.)  Components of a Cell*

| Component | Description |
| --- | --- |
| Node Agent | Each node has a Node Agent, which is a service that is used to communicate with the Deployment Manager. |
| Application Server | A node can contain standalone Application Servers (servers that are not part of a cluster). The Administration Server is an Application Server on which the Administration Console application resides. Other applications can also be targeted to the Administration Server, although this is typically done only in a single-server IBM WebSphere AS environment. |
| Cluster | A cluster typically contains Application Servers that span multiple nodes for high availability. You must create an Application Server on one of your nodes before you can create a cluster. |
| Cluster members | When creating a cluster, you add the initial cluster member (which must be an existing Application Server). After creating the cluster, you add additional cluster members to it. Each additional cluster member inherits the attributes of the first cluster member. |

*Figure 1–1   IBM WebSphere ND Cell Topology*
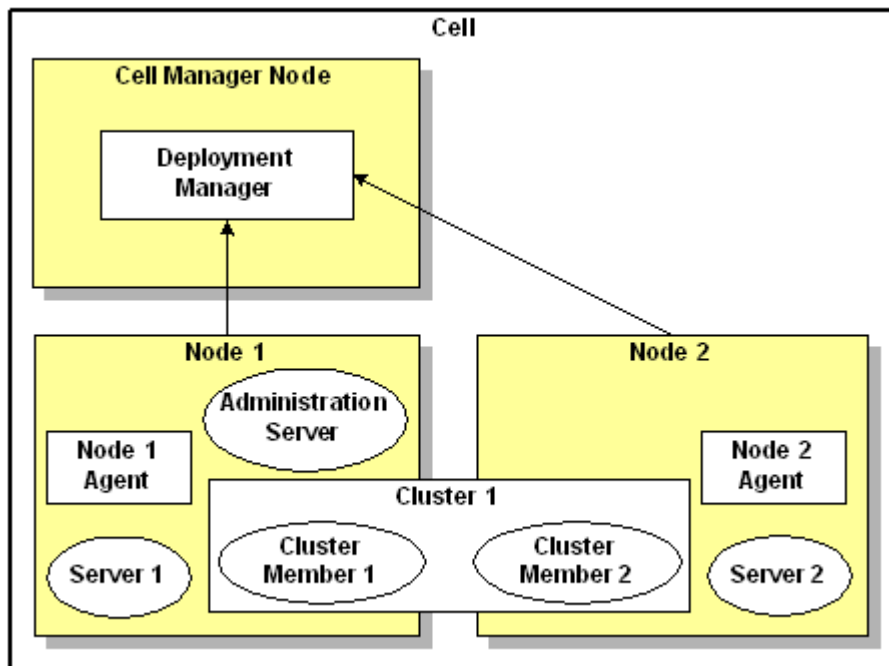


A typical Oracle Fusion Middleware product environment consists of a cell with Oracle Fusion Middleware Application Servers (including one Administration Server) residing on multiple nodes. The Cluster Member 1 and Cluster Member 2 Application Servers are clustered, with on server one Node 1 and the other server on Node 2, allowing for high availability and scalability. Server 1 and Server 2 are not clustered. To scale the environment, you add additional nodes to the cell by federating a machine to the cell, and then add new servers on those nodes.

## 1.2 Overview of the Oracle Fusion Middleware Configuration Wizard

Before you can use Oracle Fusion Middleware products in an IBM WebSphere environment, you must create and configure the cell in which the Oracle Fusion Middleware applications will run. The Oracle Fusion Middleware Configuration Wizard for IBM WebSphere simplifies the process of creating and expanding a cell for your Oracle Fusion Middleware environment. Using the wizard, you can:

- Create new IBM WebSphere cells

- Add Oracle Fusion Middleware products to the cell

- Modify an existing cell

- Add additional nodes to a cell

- Add Application Servers and clusters to a cell

- Configure the resources in a cell

The Configuration Wizard also provides the option to modify the basic configuration that is created by the product template. For example, you can add Application Servers and clusters, change JDBC data source settings, and change deployment targets.

### 1.2.1 Oracle Fusion Middleware Product Templates

The Configuration Wizard uses product templates to make it easy to add the necessary resources for an Oracle Fusion Middleware product to a cell, either when you are creating a new cell, modifying an existing cell, or federating a machine to a cell. These templates, which are Java archive (JAR) files that contain the files and scripts needed to add products to a cell, are automatically used when you select the Oracle Fusion Middleware products to include in the cell on the Add Products to Cell screen of the Configuration Wizard.

> **Caution:** Do not modify the template JAR file contents in any way. Doing so can potentially create issues in your Oracle Fusion Middleware installation.

> **Note:** `wsadmin`, which is a command line tool that you can use to configure a cell, uses the same templates as the Configuration Wizard. See Chapter 4, "Using wsadmin to Configure Oracle Fusion Middleware," for more information about `wsadmin`.

All cells that you create include the Oracle CIE Core template, which installs the Oracle CIE Custom Service and targets it to the Administration Server in the cell. The name of the Administration Server is defined as OracleAdminServer. This template is required in an IBM WebSphere environment.

When you select other products, the basic resources required by those products are also added to the cell. These may include:

- Application Servers

- Applications

- Libraries

- JDBC component schemas

- JDBC data sources

- JMS resources, such as connection factories, topics, and queues

- Services

- Custom services

- Resource adapters

The product template for a particular product defines which resources are automatically targeted to the Application Servers defined in the template, as well as the resources that are targeted to the Administration Server and Deployment Manager.

For more information about Oracle Fusion Middleware templates, see "Oracle Fusion Middleware Product Templates" in the *Oracle Fusion Middleware Domain Template Reference*, which describes the templates for WebLogic Server environments.

## 1.2.2 Template Information Specific to IBM WebSphere

Note the following differences between the IBM WebSphere environment and the WebLogic Server environment when you reference the *Oracle Fusion Middleware Domain Template Reference*:

- The Oracle CIE Core template is unique to the IBM WebSphere environment, and is required for running Oracle Fusion Middleware applications in this environment. Because this template is required, it is already selected on the Add Products to Cell screen, and cannot be deselected. It installs the Oracle CIE Custom Service and targets it to the Administration Server (OracleAdminServer) in the cell.

- The JAR file names for the IBM WebSphere versions of the templates are identical to the file names for the WebLogic Server versions.

- The template JAR files for IBM WebSphere are stored in the same path as the WebLogic Server template JAR files, except that the `applications` level of the directory path is `was`. For example, if the WebLogic Server JAR file is stored in *ORACLE_COMMON_HOME*/`common/templates/applications`, the associated JAR file for IBM WebSphere is stored in *ORACLE_COMMON_HOME*/`common/templates/was` instead.

- Some of the templates that are available for WebLogic Server are not available in an IBM WebSphere environment. To determine which templates are available for a particular product installation, see the product list on the Add Products to Cell screen of the Configuration Wizard. This list differs depending on the Oracle Fusion Middleware products that you installed.

- The product name that is displayed on the Add Products to Cell screen of the Configuration Wizard may include "for IBM WebSphere."

- In IBM WebSphere environments, the `startscript.xml` file is ignored.

## 1.2.3 Configuration Wizard Log Files

Each time that you run the Configuration Wizard, a log file is created for that session in the *MW_HOME*/`logs` directory. Log file names are in the format:

`wasconfig_`*yyyymmddhhmmss*`.log`

## 1.3 Cell Creation Process

The Configuration Wizard guides you through the process of creating a cell for your Oracle Fusion Middleware environment by enabling you to select the product components to include in the cell. If necessary, you can also customize the cell to suit your environment by:

- Customizing predefined JDBC and JMS resources

- Adding Application Servers, clusters, and cluster members

- Targeting application deployments and services to additional Application Servers or clusters

- Targeting custom services to additional Application Servers

- Targeting resource adapters to additional Application Servers

- Targeting JDBC and JMS resources to additional Application Servers or clusters

- Adding remote nodes to the cell (federating machines to a cell)

Figure 1–2 illustrates the workflow for creating a new cell. The steps indicated in boxes with dashed lines are relevant only when you create cells that contain specific products, such as those that use JDBC data sources or JMS resources.

**Figure 1–2   Workflow for Creating a New IBM WebSphere ND Cell**
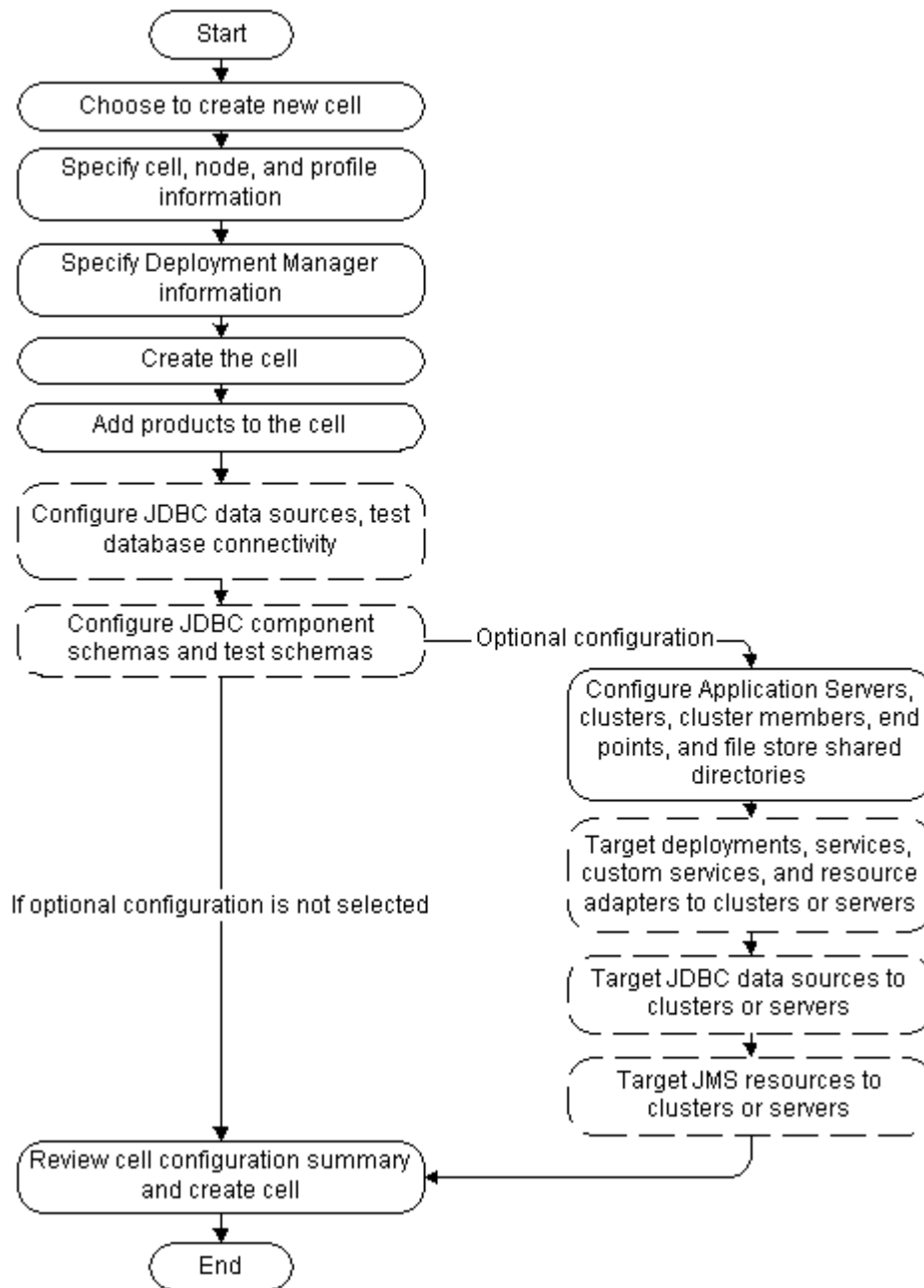


When configuring a new cell for an Oracle Fusion Middleware product environment, you can either add products to the machine on which the Deployment Manager resides or add products only on your remote nodes.

In the first case, you add the products to the cell when creating the cell, and do not have to add them when federating remote nodes to the cell.

In the second case, when you create the cell, do not add products (other than the required CIE Core product), Application Servers, and clusters to the cell at that time. Instead, use the federation process to add remote nodes to the cell, add the Oracle Fusion Middleware products to the cell, and configure your Application Servers, clusters, cluster members, and targets on the remote nodes.

## 1.4  Federation Process

You can add remote nodes to an existing IBM WebSphere ND cell by using the federation process, which is initiated by selecting the **Federate Machine and Configure Cell** option on the Select Configuration Option screen of the Configuration Wizard. After adding a remote node, you can add additional Application Servers to the cell or you can add cluster members on the node to expand an existing cluster.

Figure 1–3 illustrates the workflow for federating a machine to an existing cell on the Deployment Manager node. The steps indicated in boxes with dashed lines are relevant only when you create cells that contain specific products, such as those that use JDBC data sources or JMS resources.

**Figure 1–3   Workflow for Federating a Machine**
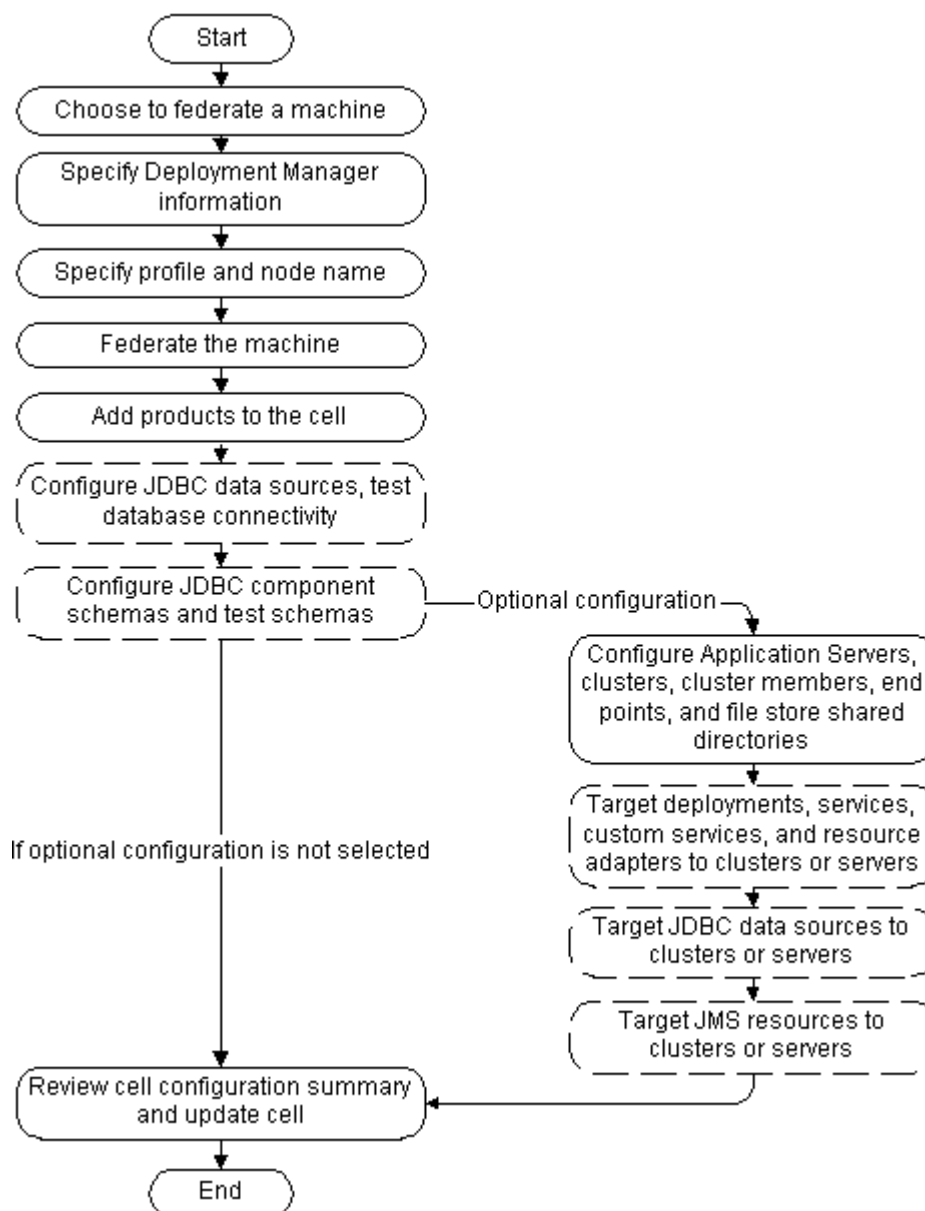
## 1.5 Creating Clusters and Cloning Application Servers

In an IBM WebSphere cell, you can create new Application Servers that inherit the attributes of an existing server by using one of the following processes:

- Clone an existing standalone Application Server to create a second standalone Application Server, which inherits its attributes from the first server.

  In this case, you clone the Application Servers by using the **Clone** function on the Configure Application Servers screen.

- Create a new cluster by using an existing Application Server as the first cluster member, and then add a new cluster member to the cluster. The new cluster member inherits the attributes of the first cluster member.

  In this case, you add the new cluster members by using the **Add** function on the Configure Additional Cluster Members screen. The new cluster member is a clone of the first cluster member.

When you create a standalone Application Server that is a clone of an existing Application Server, all applications and libraries that are targeted to the source server are also deployed to the clone. In addition, any of the following services that are targeted to the source server are automatically targeted to the clone:

- connectionFactory
- queueConnectionFactory
- topicConnection
- Queue
- Topic
- activationSpec
- Data source
- URLProvider
- workManager
- busMember
- customService
- resourceAdapter

When you create a cluster, all applications and libraries that are targeted to the first cluster member are automatically moved from that server to the cluster. In addition, any of the following services that are targeted to the first cluster member are moved from the server to the cluster:

- connectionFactory
- queueConnectionFactory
- topicConnection
- Queue
- Topic
- activationSpec
- Data source
- URLProvider

- workManager
- busMember

However, when you create a cluster, the following services are not moved from the first cluster member to the cluster. These services cannot be targeted to clusters, and can be targeted only to individual servers:

- customService
- resourceAdapter

# 2

# Using the Configuration Wizard

This chapter describes how to use the Oracle Fusion Middleware Configuration Wizard to create and update cells. It includes the following sections:

- Before You Begin
- Starting the Configuration Wizard
- Creating a New Cell
- Federating a Remote Machine to an Existing Cell
- Updating an Existing Cell
- Postconfiguration Steps
- Example Oracle Fusion Middleware Configuration
- Finding Profile Information

## 2.1 Before You Begin

If you are using a DB2 database, prior to starting the Configuration Wizard for the first time to configure FMW products, set the `DB_DRIVER_CLASSPATH` environment variable to include the full path to both `db2jcc4.jar` and `db2jcc_license_cu.jar`. If you do not do this, all DB2 connection tests will fail.

## 2.2 Starting the Configuration Wizard

The console for the machine on which the product installation resides must support Java-based graphical user interfaces (GUIs). All Windows-based consoles support Java-based GUIs; only a subset of UNIX-based consoles support Java-based GUIs.

> **Note:** If your system does not support a graphical display, you must use the `wsadmin` command to create and modify cells, and to federate your machines. See Chapter 4, "Using wsadmin to Configure Oracle Fusion Middleware," and Chapter 5, "wsadmin Command Reference," for more information.

You can start the Configuration Wizard in graphical mode from either the Windows **Start** menu or from the command line.

- To start the Configuration Wizard in graphical mode from a Windows command prompt or a UNIX command shell:

  1. Log in to the system on which the product is installed.

2. Open an MS-DOS command prompt window (on Windows) or a command shell (on UNIX).

> **Note:** On some Windows operating systems, if you have User Account Control (UAC) enabled, you may have to start the command window using the Run as Administrator command. Otherwise, you may not be able to create a cell or federate a node.

3. Go to the following directory:

   – **Windows:** *ORACLE_HOME*\common\bin

   – **UNIX:** *ORACLE_HOME*/common/bin

   Replace *ORACLE_HOME* with the directory in which you installed your Oracle Fusion Middleware products.

4. Execute the following command:

   – **Windows:** was_config.cmd

   – **UNIX:** ./was_config.sh

> **Note:** To create a log file of the Configuration Wizard session, include the -log=config.log -priority=debug parameter in the command. You can specify any file name for the log file, such as config_today.log. The log file is stored in the *logs* directory of the Oracle Middleware home directory.

## 2.3 Creating a New Cell

To create a new cell for your Oracle Fusion Middleware product installation by using the Configuration Wizard, start the Configuration Wizard as described in Section 2.2, "Starting the Configuration Wizard."

> **Note:** If you cannot run the Configuration Wizard in GUI mode, Oracle recommends that you use a wsadmin script to create or reconfigure a cell. For more information, see Chapter 4, "Using wsadmin to Configure Oracle Fusion Middleware," and Chapter 5, "wsadmin Command Reference."

The Configuration Wizard displays a sequence of screens, in the order listed in Table 2–1. For more information about each screen, see the related section in Chapter 3, "Configuration Wizard Screens," or click the link in the **Screen** column.

> **Note:** The IBM WebSphere installation that you chose when installing your Oracle Fusion Middleware products persists in the Configuration Wizard. As a result, the Welcome screen is not usually displayed, and the Select Configuration Option screen appears first.

*Table 2–1    Configuration Screens for Creating a New WebSphere Cell*

| Screen | When Does This Screen Appear? | Perform the Following Action |
| --- | --- | --- |
| Welcome | Only if the current IBM WebSphere installation is invalid, or IBM WebSphere was uninstalled after you installed your Oracle Fusion Middleware products | Click **Browse** to select the appropriate IBM WebSphere home directory for your Oracle Fusion Middleware product installation. Click **Next** to continue. |
| Select Configuration Option | Always | Select **Create and Configure Cell**. Click **Next** to continue. |
| Specify Cell, Profile and Node Name Information | Always **Note:**  The fields that you see differ depending on whether you are using IBM WebSphere AS or IBM WebSphere ND. | Fill in each field. All fields are required. Click **Next** to continue. |
| Specify Deployment Manager Information See Specify Administrator Information | Only in an IBM WebSphere ND environment | Fill in each field. All fields are required. Click **Next** to continue. |
| Specify Admin User Information See Specify Administrator Information | Only in an IBM WebSphere AS environment | Fill in each field. All fields are required. Click **Next** to continue. |
| Specify Profile and Node Name Information | Only in an IBM WebSphere AS environment | Fill in each field. All fields are required. Click **Next** to continue. |
| Creating Cell See Cell Progress | Always | Wait while the Configuration Wizard creates the new cell. When the message "Cell information read successfully, click Next to continue" is displayed, click **Next**. |
| Add Products to Cell | Always | Select the check box for each Oracle Fusion Middleware product that you want to include in the cell. Click **Next** to continue. |
| Configure the Application Server | Only if you are using IBM WebSphere AS. | Specify whether the Application Server will run in development mode or production mode. Click **Next** to continue. |
| Configure End Points | Only if you are using IBM WebSphere AS. For IBM WebSphere ND, this screen appears during optional configuration of servers and clusters | Available end points are assigned by default. You can change an end point value if you want. All end points must be unique in the cell. Click **Next** to continue. |

*Table 2–1  (Cont.)  Configuration Screens for Creating a New WebSphere Cell*

| Screen | When Does This Screen Appear? | Perform the Following Action |
|---|---|---|
| Data source configuration | Only if the cell includes components that require JDBC data sources or Oracle Real Application Cluster (Oracle RAC) multi data sources | See Section 2.3.1, "Configuring Data Sources," for information about these screens. |
| Schema configuration | Only if the cell includes components (such as Oracle SOA Suite) that require the use of the Oracle Fusion Middleware Repository Creation Utility (RCU) to load database schemas | See Section 2.3.2, "Configuring Schemas," for information about these screens. |
| Select Optional Configuration | Always | Select the check box for each category (if any) for which you want to perform advanced configuration tasks. |
| | | The available check boxes depend on the resources that are configured in the selected products or template. |
| | | Click **Next** to continue. |
| | | See Section 2.3.3, "Optional Configuration," for information about the screens that are displayed when you select options on this screen. |
| Configuration Summary | Always | Review the configuration for the cell by selecting an item from the **Summary View** list, and then selecting individual items in the list for that view. |
| | | If the cell is configured as you want it, click **Create** to create the domain. |
| | | To make changes to the configuration, click **Previous** to return to the appropriate screen for the settings that you want to change. |
| Configuring Cell | Always | This screen shows the progress of the cell creation. |
| | | When the process completes, click **Done**. |

## 2.3.1 Configuring Data Sources

The screens described in this section are displayed in the Configuration Wizard only if the domain includes components that require JDBC data sources or Oracle RAC multi data sources. Table 2–2 lists the sequences of schema-related screens. For more information about each screen, see the related section in Chapter 3, "Configuration Wizard Screens," or click the link in the **Screen** column.

*Table 2–2    Data Source Configuration Screens*

| Screen | When Does This Screen Appear? | Perform the Following Action |
|---|---|---|
| Configure JDBC Data Sources | Only if the cell includes components that require JDBC data sources | Select one or more data sources to configure, and then do one of the following:<br><br>■ Configure the fields for the data sources.<br><br>■ Select the **Configure selected data sources as RAC data sources** option.<br><br>Click **Next** to continue. |
| Configure RAC Data Sources | Only if the cell includes components that require JDBC data sources, and you selected the **Configure selected data sources as RAC data sources** option for at least one data source on the Configure JDBC Data Sources screen | Configure the fields for the Oracle RAC multi data source, including the Oracle RAC host name, instance name, and port.<br><br>Click **Next** to continue. |
| Test JDBC Data Sources | Only if the cell includes components that require JDBC data sources | Select the data sources to test, and click **Test Connections**. Wait for the connection tests to complete.<br><br>**Note:** To test connections, the database to which you are trying to connect must be running.<br><br>If you do not want to test connections at this time, deselect all data sources.<br><br>Click **Next** to continue. |

## 2.3.2  Configuring Schemas

The screens described in this section are displayed in the Configuration Wizard only if an Oracle Fusion Middleware product that you selected includes components (such as Oracle SOA Suite) that require the use of the Oracle Fusion Middleware Repository Creation Utility (RCU) to load database schemas. Table 2–3 lists the sequences of schema-related screens. For more information about each screen, see the related section in Chapter 3, "Configuration Wizard Screens," or click the link in the **Screen** column.

***Table 2–3    Schema Configuration Screens***

| Screen | When Does This Screen Appear? | Perform the Following Action |
| --- | --- | --- |
| Configure JDBC Component Schema | Only if the cell includes components (such as Oracle SOA Suite) that require the use of the Oracle Fusion Middleware Repository Creation Utility (RCU) to load database schemas | Configure the schema owner and password for each component schema listed on this screen. Changes to any of the fields on this screen are applied to all selected component schemas in the table.<br><br>For example, if all of your schemas reside on the same database, select all of the schemas in the table, and then specify the appropriate database values for the schemas (DBMS/Service, Host Name, and Port).<br><br>If, for example, you have a different password for each schema, then you must select each schema individually and specify the password for the selected schema only.<br><br>Click **Next** to continue. |
| Configure RAC Data Source Component Schema | Only if the cell includes components (such as Oracle SOA Suite) that require the use of the Oracle Fusion Middleware Repository Creation Utility (RCU) to load database schemas, and you selected the **Configured selected component schemas as RAC data source schemas** option | Configure the schema owner and password for each component schema listed on this screen. Changes to any of the fields on this screen are applied to all selected component schemas in the table.<br><br>For example, if all of your schemas reside on the same database, select all of the schemas in the table, and then specify the appropriate database values for the schemas (Service Name, Host Name, Instance Name, and Port).<br><br>If, for example, you have a different password for each schema, then you must select each schema individually and specify the password for the selected schema only.<br><br>Click **Next** to continue. |
| Test Component Schema | Only if the cell includes components (such as Oracle SOA Suite) that require the use of the Oracle Fusion Middleware Repository Creation Utility (RCU) to load a database schema | Test the configurations that you specified for the data sources in the previous screen.<br><br>Click **Next** to continue. |

## 2.3.3  Optional Configuration

When creating a new cell, updating an existing cell, or federating a machine to a cell, you can:

- Add Application Servers, clusters, and cluster members to the cell

- Target application deployments and services to additional Application Servers or clusters in the cell

- Target custom services to additional Application Servers

- Target JDBC and JMS resources to additional Application Servers or clusters

Table 2–4 lists the sequence of screens that is displayed in the Configuration Wizard when you select one or more configuration options on the Select Optional

Configuration screen. The screens that are displayed depend on the options that you selected.

*Table 2–4    Optional Configuration Screens*

| Screen | When Does This Screen Appear? | Perform This Action |
|---|---|---|
| Configure Application Servers | Only if you are using IBM WebSphere ND, and you selected **Application Servers, Clusters, and End Points** on the Select Optional Configuration screen | Add, delete, or configure Application Servers as needed.<br><br>Click **Next** to continue. |
| Configure Clusters | Only if you are using IBM WebSphere ND, and you selected **Application Servers, Clusters, and End Points** on the Select Optional Configuration screen | Add or delete clusters as needed. When adding a cluster, enter the cluster name and select the first cluster member from the list.<br><br>Click **Next** to continue. |
| Configure Additional Cluster Members | Only if you are using IBM WebSphere ND, and you selected **Application Servers, Clusters, and End Points** on the Select Optional Configuration screen, and you created clusters on the Configure Clusters screen | Add or delete cluster members as needed.<br><br>Click **Next** to continue. |
| Configure End Points | Only if you are using IBM WebSphere ND, and you selected **Application Servers, Clusters, and End Points** on the Select Optional Configuration screen | Unique end points are assigned by default. You can change an end point on a server if you want. All end points must be unique in the cell.<br><br>Click **Next** to continue. |
| Configure File Store Shared Directories | Only if:<br><br>■ You are using IBM WebSphere ND<br><br>■ You selected **Application Servers, Clusters, and End Points** on the Select Optional Configuration screen<br><br>■ A selected product defines servers that use a file store<br><br>■ You have created a cluster that includes a server that uses a file store | For each directory (log, permanent store, or temporary store) you want to change, enter the new directory name in the appropriate column and row.<br><br>Click **Next** to continue. |

*Table 2–4 (Cont.) Optional Configuration Screens*

| Screen | When Does This Screen Appear? | Perform This Action |
|---|---|---|
| Target Deployments to Clusters or Servers | Only if you selected **Deployments and Services** on the Select Optional Configuration screen, and a selected product contains J2EE applications or libraries | Select the target server or cluster on which you want to deploy applications or libraries.<br><br>Select the check box for each application or library that you want to deploy on the selected target.<br><br>Click **Next** to continue. |
| Target Services to Clusters or Servers | Only if you selected **Deployments and Services** on the Select Optional Configuration screen | Select the target server or cluster.<br><br>Select the check box for each service that you want to target to the server or cluster.<br><br>Click **Next** to continue. |
| Target Custom Services to Servers | Only if you selected **Deployments and Services** on the Select Optional Configuration screen and your environment contains custom services | Select the target server.<br><br>Select the check box for each service that you want to target to the server.<br><br>Click **Next** to continue. |
| Target Resource Adapters to Servers | Only if you selected **Deployments and Services** on the Select Optional Configuration screen and your product environment contains resource adapters | Select the target server.<br><br>Select the check box for each resource adapter that you want to target to the server.<br><br>Click **Next** to continue. |
| Target JDBC Data Sources to Clusters or Servers | Only if you selected **JDBC** on the Select Optional Configuration screen | Select the target server or cluster.<br><br>Select the check box for each JDBC data source that you want to target to the server or cluster.<br><br>Click **Next** to continue. |
| Target JMS Resources to Clusters or Servers | Only if you selected **JMS** on the Select Optional Configuration screen | Select the target server or cluster.<br><br>Select the check box for each JMS resource that you want to target to the server or cluster.<br><br>Click **Next** to continue. |

## 2.4 Federating a Remote Machine to an Existing Cell

In an IBM WebSphere ND environment, you can create Application Server nodes on one or more remote machines, and associate those nodes with the Deployment Manager that will manage them. This process is called federating a machine. It is the process that you use to add new nodes to a cell that you previously created from the Deployment Manager node.

> **Note:** Prior to federating a machine, you must install IBM WebSphere ND and your Oracle Fusion Middleware products on that machine. The IBM WebSphere ND and Oracle Fusion Middleware product versions must be identical on all machines.

After adding remote nodes to an existing cell on the Deployment Manager host, you can:

- Add standalone Application Servers on each node to the cell
- Create clusters that include multiple Application Servers on each node, or that contain Application Servers on multiple nodes
- Deploy applications on each Application Server or cluster
- Target services to each Application Server or cluster
- Target custom services to each Application Server
- Target resource adapters to each Application Server
- Target JDBC and JMS resources to each Application Server or cluster

To federate a remote machine to an existing cell on the Deployment Manager node:

1. If you have not already created the cell to which the new node will belong, follow the steps in Section 2.3, "Creating a New Cell," to create a new cell on the Deployment Manager node.

   The cell does not need to contain any products or be otherwise configured; you only need to create the Deployment Manager profile and an empty custom profile. Typically, you will want to run the Oracle Fusion Middleware applications on remote nodes and not run them on the Deployment Manager node. Therefore, you should add products during the federation process or by editing the cell configuration later.

2. On the Deployment Manager node, start the Deployment Manager as follows:

   If only one Deployment Manager is configured on the machine or if you are federating the remote node to a cell that is managed by the default Deployment Manager, execute the following commands, where `WAS_HOME` is the IBM WebSphere ND home directory:

   - **UNIX:**

     ```
     cd WAS_HOME/bin

     ./startManager.sh
     ```

   - **Windows:**

     ```
     cd WAS_HOME\bin

     startManager.cmd
     ```

   If multiple Deployment Managers are configured on the machine and you want to start a Deployment Manager other than the default, execute the following commands, where `WAS_HOME` is the IBM WebSphere ND home directory, and `DepMgr` is the name of the Deployment Manager that will manage the remote nodes (for example, `Dmgr01`):

   - **UNIX:**

     ```
     cd WAS_HOME/profiles/DepMgr/bin

     ./startManager.sh
     ```

   - **Windows:**

     ```
     cd WAS_HOME\profiles\DepMgr\bin

     startManager.cmd
     ```

3. On the remote machine, start the Configuration Wizard as follows, where `ORACLE_HOME` is the directory in which you installed your Oracle Fusion Middleware products:

- **UNIX:**

  cd *ORACLE_HOME*/common/bin

  ./was_config.sh

- **Windows:**

  cd *ORACLE_HOME*\common\bin

  was_config.cmd

**4.** See Table 2–5 for the sequence of screens that are displayed in the Configuration Wizard when you are federating a machine. For more information about each screen, see the related section in Chapter 3, "Configuration Wizard Screens," or click the link in the **Screen** column.

*Table 2–5    Configuration Screens for Federating a Machine*

| Screen | When Does This Screen Appear? | Perform the Following Action |
|---|---|---|
| Welcome | Only if the current IBM WebSphere installation is invalid, or IBM WebSphere was uninstalled after you installed your Oracle Fusion Middleware products | Click **Browse** to select the appropriate IBM WebSphere home directory. Click **Next** to continue. |
| Select Configuration Option | Always | Select **Federate Machine and Configure Cell**. Click **Next** to continue. |
| Specify Administrator Information | Always | Specify the appropriate information for the Deployment Manager node. Click **Next** to continue. |
| Specify Profile and Node Name Information | Always | Specify the profile name and node for the Application Server node that you are creating. Click **Next** to continue. |
| Federating Machine (see Cell Progress) | Always | Displays the progress of the federation process. When the message "Cell information read successfully, please click Next" is displayed, click **Next** to continue. |
| Add Products to Cell | Always | Select the check box for each Oracle Fusion Middleware product that you want to add to this cell. Click **Next** to continue. |
| Data source configuration | Only if the cell includes components that require JDBC data sources or Oracle RAC data sources | See Section 2.3.1, "Configuring Data Sources," for information about these screens. |

*Table 2–5   (Cont.)  Configuration Screens for Federating a Machine*

| Screen | When Does This Screen Appear? | Perform the Following Action |
| --- | --- | --- |
| Schema configuration | Only if the cell includes components (such as Oracle SOA Suite) that require the use of the Oracle Fusion Middleware Repository Creation Utility (RCU) to load database schemas | See Section 2.3.2, "Configuring Schemas," for information about these screens. |
| Select Optional Configuration | Always | Select the check box for each category (if any) for which you want to perform advanced configuration tasks. |
| | | The available check boxes depend on the resources that are configured in the selected products or template. |
| | | Click **Next** to continue. |
| | | See Section 2.3.3, "Optional Configuration," for information about the screens that are displayed when you select options on this screen. |
| Configuration Summary | Always | Review the configuration for the cell by selecting a **Summary View**, and then selecting individual items in the list for that view. |
| | | If the cell is configured as you want it, click **Extend** to add the remote node to the cell. |
| | | To make changes to the configuration, click **Previous** to return to the appropriate screen for the settings you want to change. |
| Configuring Cell | Always | Shows the progress of the cell update. |
| | | When the process completes, click **Done**. |

## 2.5  Updating an Existing Cell

To update an existing cell for your Oracle Fusion Middleware product installation by using the Configuration Wizard, start the Configuration Wizard as described in Section 2.2, "Starting the Configuration Wizard." Oracle recommends that you run the Configuration Wizard from the Deployment Manager machine. If you run the Configuration Wizard from a remote machine, ensure that the Deployment Manager is running.

The Configuration Wizard displays a sequence of screens, in the order listed in Table 2–6. For more information about each screen, see the related section in Chapter 3, "Configuration Wizard Screens," or click the link in the **Screen** column.

***Table 2–6    Configuration Screens for Editing an Existing Cell***

| Screen | When Does This Screen Appear? | Perform the Following Action |
|---|---|---|
| Welcome | Only if the current IBM WebSphere installation is invalid, or IBM WebSphere was uninstalled after you installed your Oracle Fusion Middleware products. | Click **Browse** to select the appropriate IBM WebSphere home directory.<br><br>Click **Next** to continue. |
| Select Configuration Option | Always | Select **Select and Configure Existing Cell**.<br><br>From the list of cells displayed under this option, click the cell that you want to edit. The selected cell is highlighted.<br><br>Click **Next** to continue. |
| Loading Cell Information<br><br>(see Cell Progress) | Always | Wait while the Configuration Wizard loads the selected cell.<br><br>When the message "Cell information read successfully, please click Next" is displayed, click **Next** to continue. |
| Add Products to Cell | Always | Select the check box for each Oracle Fusion Middleware product that you want to add to this cell.<br><br>Click **Next** to continue. |
| Data source configuration | Only if the cell includes components that require JDBC data sources or Oracle RAC data sources | See Section 2.3.1, "Configuring Data Sources," for information about these screens. |
| Schema configuration | Only if the cell includes components (such as Oracle SOA Suite) that require the use of the Oracle Fusion Middleware Repository Creation Utility (RCU) to load database schemas | See Section 2.3.2, "Configuring Schemas," for information about these screens. |
| Select Optional Configuration | Always | Select the check box for each category (if any) for which you want to perform advanced configuration tasks.<br><br>The available check boxes depend on the resources that are configured in the selected products or template.<br><br>Click **Next** to continue.<br><br>See Section 2.3.3, "Optional Configuration," for information about the screens that are displayed when you select options on this screen. |

*Table 2–6   (Cont.)  Configuration Screens for Editing an Existing Cell*

| Screen | When Does This Screen Appear? | Perform the Following Action |
|---|---|---|
| Configuration Summary | Always | Review the configuration for the cell by selecting a **Summary View**, and then selecting individual items in the list for that view. |
| | | If the cell is configured as you want it, click **Extend** to save the new cell configuration. |
| | | To make changes to the configuration, click **Previous** to return to the appropriate screen for the settings that you want to change. |
| Configuring Cell | Always | This page shows the progress of the cell configuration update. |
| | | When the process completes, click **Done**. |

## 2.6  Postconfiguration Steps

After completing the cell configuration, perform the steps in the following sections to:

- Start the Deployment Manager for the cell

- Start the Node Agent for the Application node on the Deployment Manager machine

- Start the Administration Server for the cell

- Start the Application Servers in the cell

- Start the Node Agent on each remote node in the cell

**Deployment Manager Machine**

Perform the following steps on the Deployment Manager machine to start the Deployment Manager, Node Agent, and Administration Server:

**1.** If the Deployment Manager is not already running, start the Deployment Manager as follows:

If only one Deployment Manager is configured on the machine or to start the default Deployment Manager, execute the following commands, where *WAS_HOME* is the IBM WebSphere ND home directory:

- **UNIX:**

  cd *WAS_HOME*/bin

  ./startManager.sh

- **Windows:**

  cd *WAS_HOME*\bin

  startManager.cmd

If multiple Deployment Managers are configured on the machine and you want to start a Deployment Manager other than the default, execute the following commands, where *WAS_HOME* is the IBM WebSphere ND home directory, and *DepMgrProfile* is the profile name of the Deployment Manager that you want to start (for example Dmgr02):

- **UNIX:**

```
cd WAS_HOME/profiles/DepMgrProfile/bin

./startManager.sh
```

- **Windows:**

```
cd WAS_HOME\profiles\DepMgrProfile\bin

startManager.cmd
```

2. Enter the following commands to start the Node Agent, where `Profile` is the name of the Application Server profile on the Deployment Manager machine (for example, `Custom01`):

- **UNIX:**

```
cd WAS_HOME/profiles/Profile/bin

./startNode.sh -profileName Profile
```

- **Windows:**

```
cd WAS_HOME\profiles\Profile\bin

startNode.cmd -profileName Profile
```

3. Enter the following command to start the Administration Server:

- **UNIX:**

```
./startServer.sh OracleAdminServer -profileName Profile
```

- **Windows:**

```
startServer.cmd OracleAdminServer -profileName Profile
```

4. To start individual servers, enter the following command for each server:

- **UNIX:**

```
./startServer.sh server_name -profileName Profile
```

- **Windows:**

```
startServer.cmd server_name -profileName Profile
```

**Remote Nodes**

On each remote node, enter the following command to start the Node Agent on the node, where `NodeProfile` is the Application Server profile name that you configured on the node (for example, `Custom04`):

- **UNIX:**

```
cd WAS_HOME/profiles/NodeProfile/bin

./startNode.sh -profileName NodeProfile
```

- **Windows:**

```
cd WAS_HOME\profiles\NodeProfile\bin

startNode.cmd -profileName NodeProfile
```

## 2.7 Example Oracle Fusion Middleware Configuration

This section provides an example configuration of an Oracle SOA Suite installation in an IBM WebSphere ND environment. The procedure assumes that you have already

installed IBM WebSphere ND and the Oracle SOA Suite on each machine that will be included in the configuration.

In this example:

- A new cell is created on the Deployment Manager node.

- The Administration Server is configured on the Deployment Manager machine.

- The Oracle SOA Suite and Java Required Files (JRF) products are added to the cell. The Oracle SOA Suite product template automatically adds a soa_server_1 Application Server to the cell (it resides on the same machine as the Deployment Manager node).

- A soa_cluster_1 cluster is created on the Application Server node, and the soa_server_1 Application Server is added as the primary cluster member.

- A remote node is added to the cell.

- A soa_server_2 Application Server is added to the remote node as a secondary member of the soa_cluster_1 cluster.

Figure 2–1 shows the resulting topology of the cell.

*Figure 2–1   Example Configuration*
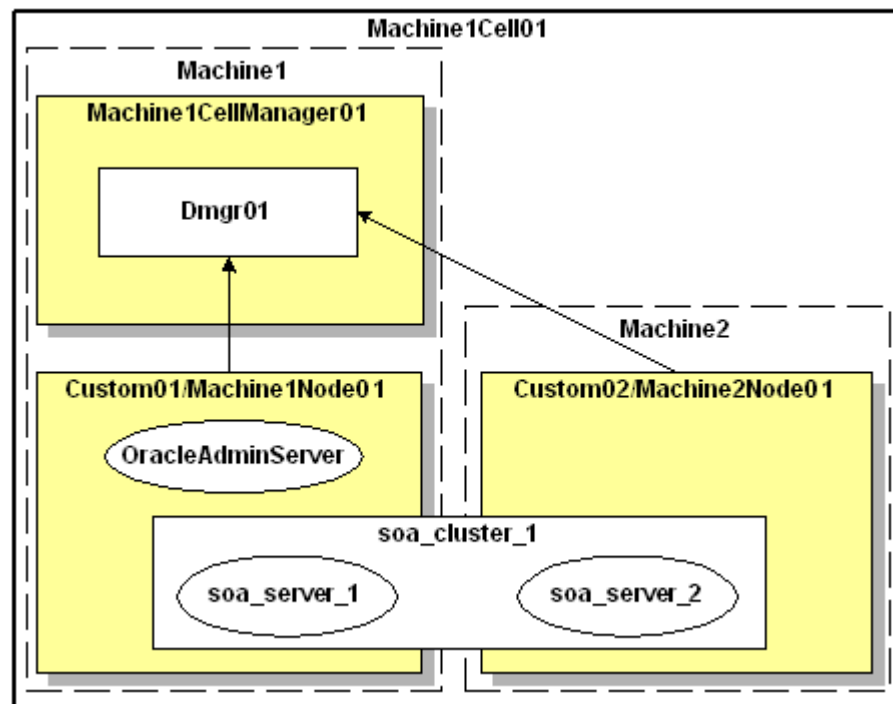


### Creating the Cell

To create this cell configuration, first perform the following steps on the Deployment Manager machine.

1. Run the Configuration Wizard as described in Section 2.2, "Starting the Configuration Wizard."

2. On the Select Configuration Option screen of the wizard, select **Create and Configure Cell** and click **Next**.

3. Specify the cell name (Machine1Cell01), Deployment Manager profile name (Dmgr01), Deployment Manager node name (Machine1CellManager01), Application Server profile name (Custom01), and Application Server node name (Machine1Node01). Click **Next**.

4. Specify the Deployment Manager host name, and create the user name and password for the Administrator account. This is the user name and password that you will use to connect to the Deployment Manager from the remote node. When you are done, click **Next**.

5. Wait while the new cell is created, and then click **Next** to continue.

6. On the Add Products screen, the **Oracle CIE Core** product is required and is already selected. Select the **Oracle SOA Suite for WebSphere ND** product. The **Oracle Workflow Client Extension**, **Oracle WSM Policy Manager**, and **Oracle JRF for WebSphere** products are then automatically selected, as they are required by SOA Suite. Click **Next** to continue.

   > **Note:** If you do not want to run Oracle Fusion Middleware applications on the Deployment Manager host, install only the **Oracle CIE Core** product in this step. When you are configuring your first remote node, add **Oracle SOA Suite** and its supporting products to the cell.

7. On the Configure JDBC Component Schema screen, select one or more component schemas and configure them appropriately. When done, click **Next**.

   > **Note:** There are no JDBC data sources defined in configuration templates for the selected products, so the Configure JDBC Data Sources and Test JDBC Data Sources screens are not displayed.

8. On the Test Component Schema screen, wait for the schema tests to run. When a check mark appears to the left of each schema, click **Next** to continue.

9. On the Select Optional Configuration screen, select **Application Servers, Clusters, and End Points**. This allows you to create the soa_cluster_1 cluster and add soa_server_1 to it. There is no need to select the **Deployments and Services JDBC**, or **JMS** options, as all deployments, services, and other resources have already been targeted to the appropriate servers (OracleAdminServer or soa_server_1 or both). Click **Next**.

   > **Note:** If you do not want to run Oracle Fusion Middleware applications on the Deployment Manager host, simply click **Next** on the Select Optional Configuration screen, and go to Step 15. Instead, create the cluster on the first remote node you add to the cell.

10. On the Configure Application Servers screen, you will see **OracleAdminServer** and **soa_server_1** in the list. Click **Next** to continue. For this example, there is no need to add additional application servers.

11. On the Configure Clusters screen, click **Add** to add a new cluster. Enter **soa_cluster_1** in the **Name** column, and select **soa_server_1** from the list in the **First cluster member** column. When you are done, click **Next**.

12. On the Add Additional Cluster Members screen, click **Next**. For this example, there is no need to add additional cluster members at this time. You will add the soa_server_2 cluster member when you configure the remote node.

13. On the Configure End Points screen, click **Next**. For this example, there is no need to change the default port values. You may want to note the SOAP end point value, as this is the port you will use to connect to the Deployment Manager machine from a remote node.

14. On the Configure File Store Shared Directories screen, click **Next**. For this example, there is no need to change the defaults.

15. On the Configuration Summary screen, click **Create** to start the cell creation process.

16. On the Configuring Cell screen, wait for the cell to be created, and then click **Done** to exit the Configuration Wizard.

17. In the command window, enter the following commands to start the Deployment Manager, where *WAS_HOME* is the IBM WebSphere ND home directory:

    cd *WAS_HOME*/profiles/Dmgr01/bin/

    - **UNIX:**

      cd *WAS_HOME*/profiles/Dmgr01/bin

       ./startManager.sh -profileName Dmgr01

    - **Windows:**

      cd *WAS_HOME*\profiles\Dmgr01\bin

      startManager.cmd -profileName Dmgr01

Continue to the next procedure to add the remote node.

### Federating a Remote Machine to the Cell

After completing the steps on the Deployment Manager machine, perform the following steps on the remote machine:

1. Connect to the remote machine and start the Configuration Wizard as described in Section 2.2, "Starting the Configuration Wizard."

2. On the Select Configuration Option screen, select **Federate Machine and Configure Cell** and click **Next**.

3. Specify a unique profile name (Custom02) and node name (*machine2_name*Node01). Click **Next**.

4. On the Specify Deployment Manager Information screen:

   - Specify the host name of the Deployment Manager machine (Dmgr01).

   - Enter the SOAP port used to connect to the Deployment Manager (the default is 8880).

   - Enter the user name and password you specified in Step 4 of the previous procedure.

   - Click **Next** to continue.

5. Wait while the node is federated to the cell.

6. On the Add Products screen, the **Oracle SOA Suite for WebSphere ND**, **Oracle Workflow Client Extension**, **Oracle WSM Policy Manager**, and **Oracle JRF for**

**WebSphere** and products are already selected, as they were added to the cell when you created it. Click **Next**.

7. On the Configure JDBC Component Schema screen, click **Next**.

8. On the Test Component Schema screen, wait for the schema tests to run. When a check mark appears to the left of each schema, click **Next** to continue.

9. On the Select Optional Configuration screen, select **Application Servers and Clusters**. This allows you to add soa_server_2 to the soa_cluster1 cluster that you already created on the first node.

10. On the Configure Application Servers screen, you will see **OracleAdminServer** and **soa_server_1** in the list. Click **Next** to continue. For this example, there is no need to add additional Application Servers.

11. On the Configure Clusters screen, you will see **soa_server_1** in the cluster list. There is no need to create another cluster. Click **Next**.

12. On the Add Additional Cluster Members screen, click **Add** to add soa_server_2 to soa_cluster_1.

13. Enter **soa_server_2** in the **Name** column, select the current node from the **Node** list, and select **soa_cluster_1** from the **Cluster** list. The new cluster member inherits all attributes from the primary cluster member. When you are done, click **Next**.

14. On the Configure End Points screen, click **Next**. For this example, there is no need to change the default port values. Unique end points are assigned to the new cluster member.

15. On the Configure File Store Shared Directories screen, click **Next**. For this example, there is no need to change the defaults.

16. On the Configuration Summary screen, click **Create** to start the cell creation process.

17. On the Configuring Cell screen, wait for the cell to be created, and then click **Done** to exit the Configuration Wizard.

To scale the cluster to include additional Application Servers on other remote nodes, repeat the steps in this section using unique profile names and server names.

## 2.8 Finding Profile Information

Key information about the Deployment Manager profile and Application Server profiles can be found in the following file:

*WAS_HOME*/profiles/*profile_name*/logs/AboutThisProfile.txt

Deployment Manager profile information includes:

- Disk space required for the profile

- Whether or not the profile is the default profile

- Cell Manager node name

- Cell name

- Host name

- Administrative Console port

- Administrative Console secure port

- Management bootstrap port

- Management SOAP connector port

- Whether or not the Deployment Manager is run as a service

Application Server profile information includes:

- Whether or not the profile is the default profile

- Application Server node name

- Application Server host name

# 3

# Configuration Wizard Screens

This chapter contains screen descriptions for all of the Oracle Fusion Middleware Configuration Wizard for WebSphere.

- Welcome
- Select Configuration Option
- Specify Cell, Profile and Node Name Information
- Specify Administrator Information
- Specify Profile and Node Name Information
- Cell Progress
- Add Products to Cell
- Configure the Application Server
- Configure JDBC Data Sources
- Configure RAC Data Sources
- Test JDBC Data Sources
- Configure JDBC Component Schema
- Configure RAC Data Source Component Schema
- Test Component Schema
- Select Optional Configuration
- Configure Application Servers
- Configure Clusters
- Configure Additional Cluster Members
- Configure End Points
- Configure File Store Shared Directories
- Target Deployments to Clusters or Servers
- Target Services to Clusters or Servers
- Target Custom Services to Servers
- Target Resource Adapters to Servers
- Target JDBC Data Sources to Clusters or Servers
- Target JMS Resources to Clusters or Servers

- Configuration Summary

- Configuring Cell

These screens contain the following standard buttons:

- **Exit**: Exit the wizard without saving configuration changes.

- **Help**: Display help for the current page. You can then navigate to other help topics if necessary.

- **Previous**: Return to the previous screen.

- **Next**: Continue to the next screen.

Some screens may also contain the following function buttons:

- **Discard Changes**: Discard any changes you made on the current screen.

- **Delete**: Delete the selected item or items.

- **Switch Display**: Toggle between a table format (one row for each item) and a tabbed format (one tab for each item).

## 3.1 Welcome
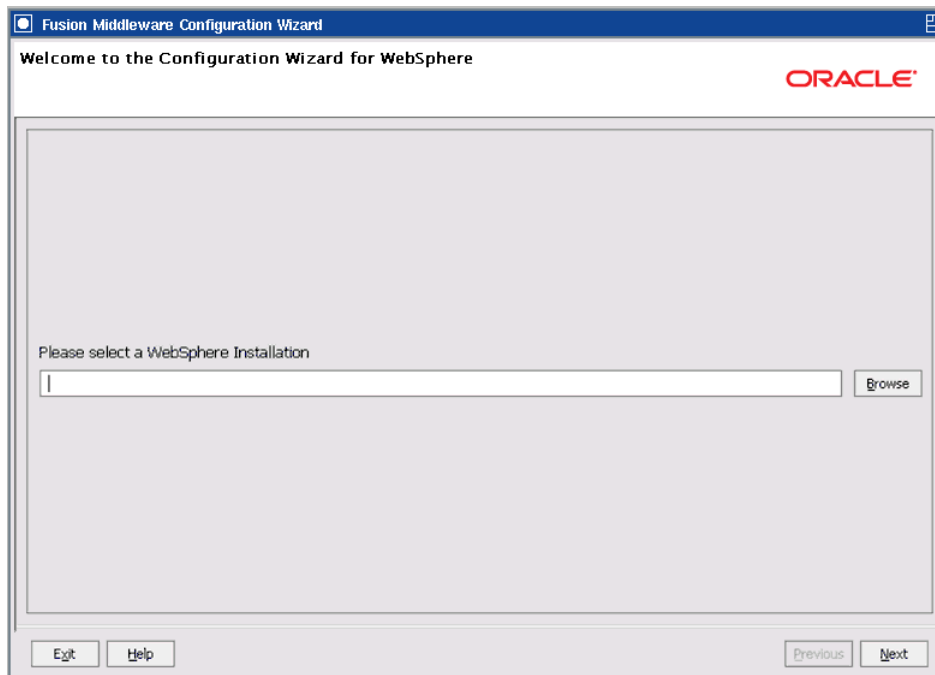


Click **Browse** to navigate to the home directory for the IBM WebSphere installation for which you want to configure a new cell, update an existing cell, or federate a machine. After navigating to the directory, click **Open**.

Click **Next** to Continue.

## 3.2 Select Configuration Option
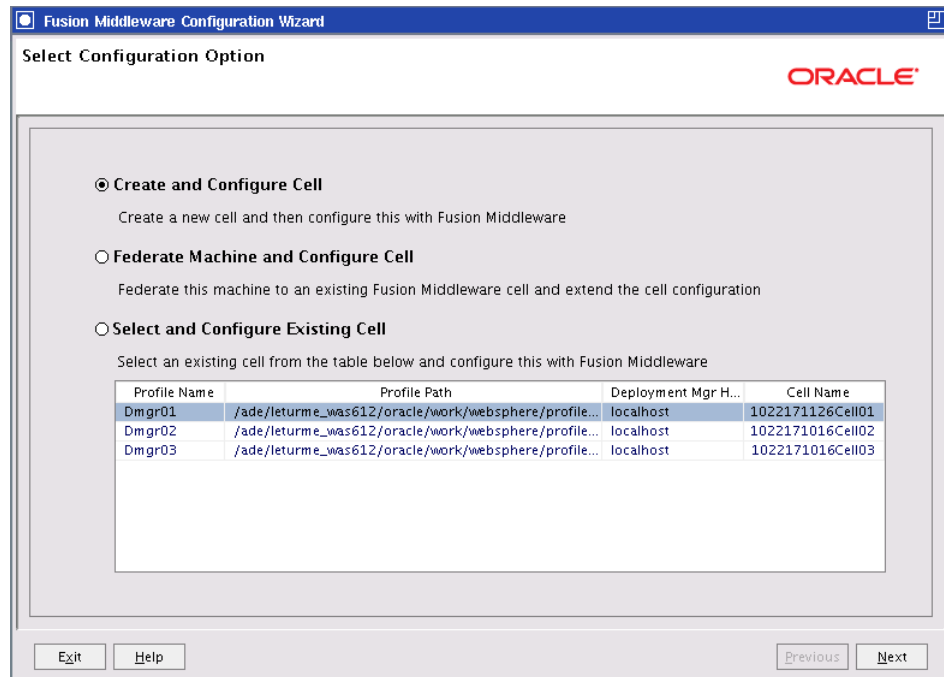


Select one of the following options. The options that you see depend on your current configuration.

- **Create and Configure Cell**: Select this option to create a new cell. The Specify Cell, Profile and Node Name Information screen appears next.

- **Federate Machine and Configure Cell**: (IBM WebSphere ND only) Select this option to add the current node to an existing cell. The Specify Profile and Node Name Information screen appears next. In this case, you must run the Configuration Wizard on the node that you want to federate to the cell.

  ---
  **Note:** This option is not displayed for IBM WebSphere AS environments, as it does not apply to single-server installations.

  ---

- **Select and Configure Existing Cell**: This option is available only if you have already created a cell. To change the configuration of an existing cell, select this option. From the list, click on the cell that you want to configure (the selected cell will be highlighted). The Loading Cell Information screen appears next. See Section 3.6, "Cell Progress," for information about this screen.

## 3.3 Specify Cell, Profile and Node Name Information

This screen differs depending on whether you are running the Configuration Wizard in an IBM WebSphere ND environment or an IBM WebSphere AS environment.

In an IBM WebSphere AS environment:

- There is no Deployment Manager profile or Deployment Manager node. Therefore, the **Deployment Manager Profile Name** and **Deployment Manager Node Name** fields are not available on this screen.

- The **Application Server Name** field is available, allowing you to assign a name to the Application Server.

The following image shows the IBM WebSphere ND version of this screen.



Enter the appropriate values in the following fields. Each field is required. Use only alphanumeric characters in each field.

- **Cell Name**: Enter a unique name for the new cell. This is the name of the directory in which the cell is created, under the following two directories:

    - **IBM WebSphere ND only**: `WAS_ HOME`/profiles/`DepMgrProfile`/config/cells, where `WAS_HOME` is the IBM WebSphere ND installation directory and `DepMgrProfile` is the name of the Deployment Manager profile.

    - **Both IBM WebSphere AS and IBM WebSphere ND**: `WAS_ HOME`/profiles/`AppSrvrProfile`/config/cells, where `WAS_HOME` is the IBM WebSphere installation directory and `AppSrvrProfile` is the name of the Application Server profile.

    The default name is *machine_name*Cell*n*, where *machine_name* is the name of the current machine, and *n* is the next two-digit number in the sequence on that machine. For example, if the machine name is machine1 and Cell01 already exists, the default cell name is machine1Cell02.

- **Deployment Manager Profile Name**: (IBM WebSphere ND only) Enter a unique Deployment Manager profile name for this cell. This is the name of the directory in which the Deployment Manager profile is created, under `WAS_HOME`/profiles.

    The default name is Dmgr*n*, where *n* is the next two-digit number in the sequence. For example, if Deployment Manager profile Dmgr01 already exists, the default for this field is Dmgr02.

- **Deployment Manager Node Name**: (IBM WebSphere ND only) Enter a unique Deployment Manager node name for this cell.

The default name is *machine_name*CellManager*n*, where *machine_name* is the name of the current machine, and *n* is the next two-digit number in the sequence. For example, if the machine name is machine1 and CellManager01 already exists, the default node name is machine1CellManager02.

- **Application Server Profile Name**: Enter a unique Application Server profile name for this cell. The is the name of the directory in which the Application Server profile is created, under *WAS_HOME*/`profiles`.

  - **IBM WebSphere ND:** The default name is Custom*n*, where *n* is the next two-digit number in the sequence. For example, if Application Server profile Custom01 already exists, the default for this field is Custom02.

  - **IBM WebSphere AS:** The default name is AppSrv01.

- **Application Server Node Name**: Enter a unique Application Server node name for this cell.

  The default name is *machine_name*Node*n*, where *machine_name* is the name of the current machine, and *n* is the next two-digit number in the sequence on the machine. For example, if the machine name is machine1 and Node01 already exists, the default node name is machine1Node02.

- **Application Server Name**: (IBM WebSphere AS only) If you have already created the Application Server, enter the name of the Application Server to use for this cell. The Configuration Wizard will designate this server internally as the Administration Server for this cell.

  If you have not already created the Application Server, the server name defaults to OracleAdminServer. The Configuration Wizard will create the Application Server using this name, and designate it internally as the Administration Server for this cell.

When you are done filling in each field, click **Next**.

## 3.4  Specify Administrator Information

In an IBM WebSphere ND environment, you specify the Administrator user name and password on the Specify Deployment Manager Information screen (shown here).

In an IBM WebSphere AS environment, you specify the Administrator user name and password on the Specify Admin User Information screen.

Use this screen to specify the host name, SOAP port (federation process only), administrator user name and password for the Deployment Manager. All fields are required.

- **Deployment Manager Host Name**: (IBM WebSphere ND only) When you are creating a new cell, this field defaults to the host name of the current machine.

  When you are federating a machine, enter the host name of the Deployment Manager that will manage the remote node.

- **Host Name**: (IBM WebSphere AS only) This field defaults to the host name of the current machine.

- **SOAP Port**: (IBM WebSphere ND only) This field is displayed only when you are federating a machine. Enter the SOAP port to use for connecting to the Deployment Manager node. This is the SOAP port of the Deployment Manager to which you want to federate this machine. The default is typically 8880. If you do not know the SOAP port for the Deployment Manager, open the following file on the Deployment Manager host:

  *WAS_HOME*/profiles/*DepMgr_name*/logs/AboutThisProfile.txt

  Replace *WAS_HOME* with the IBM WebSphere home directory and replace *DepMgr_name* with the name of the appropriate Deployment Manager profile on the specified host.

- **Admin User Name**: When creating a new cell, enter the administrator user name that you want to use to start or stop the Deployment Manager (IBM WebSphere ND) or start or stop the Administration Server (IBM WebSphere AS).

  (IBM WebSphere ND only) When you are federating a machine, enter the administrator user name that was configured for the Deployment Manager on the specified host.

- **Password**: When you are creating a new cell, enter the administrator password that you want to use for the cell.

(IBM WebSphere ND only) When you are federating a machine, enter the administrator password that was configured for the Deployment Manager on the specified host.

■ **Confirm Password**: This field is displayed only when you are creating a new cell. Reenter the administrator password to confirm it. This entry must exactly match the entry in the **Password** field.

---

**Note:** The Administrator user name and password that you enter here are used to connect to the Deployment Manager, and to stop the Deployment Manager, Node Managers, Administration Server, or Application Servers.

---

When you are done filling in each field, click **Next** to continue.

## 3.5 Specify Profile and Node Name Information



Use this screen to specify the Application Server Profile name and Node Name for an IBM Websphere AS installation. Both fields are required.

■ **Application Server Profile Name**: Enter a unique Application Server profile name for this node. This is the name of the directory in which the Application Server profile is created, under *WAS_HOME*/`profiles` on the machine that is hosting the profile.

The default name is Custom*n*, where *n* is the next two-digit number in the sequence for the current cell. For example, if Application Server profile Custom01 already exists in the cell, the default for this field is Custom02.

If you change the default profile name, ensure that it is unique within the cell.

■ **Application Server Node Name**: Enter the node name to use for this Application Server.

The default name is *machine_name*Node*n*, where *machine_name* is the name of the current machine, and *n* is the next two-digit number in the sequence on the current machine. For example, if the machine name is machine2 and Node01 already exists on the machine, the default node name is machine2Node02.

If you change the default node name, ensure that it is unique within the cell.

When you are done filling in each field, click **Next** to continue.

## 3.6 Cell Progress
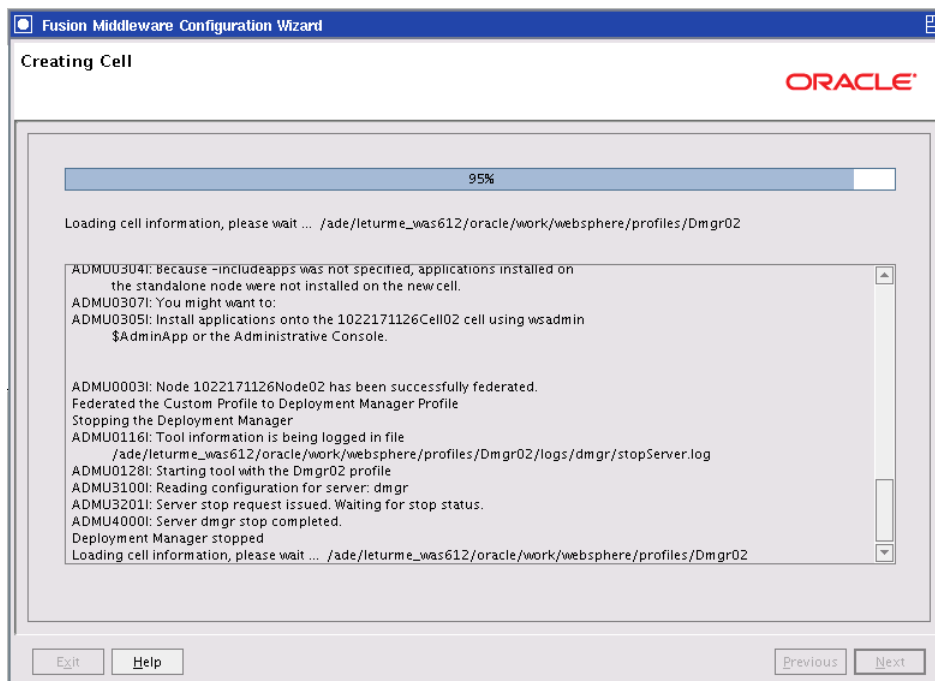


This screen displays the status of the cell creation, cell loading, or federating machine process. This example shows the Creating Cell screen. The screen title differs depending on your current task:

- When you are creating a new cell, the screen title is Creating Cell.

- When you are updating an existing cell, the screen title is Loading Cell Information.

- When you are federating a remote machine to a cell, the screen title is Federating Machine.

When the progress bar reaches 100% and the message "Cell information read successfully, please click Next" is displayed in the lower window, click **Next** to continue.

## 3.7  Add Products to Cell
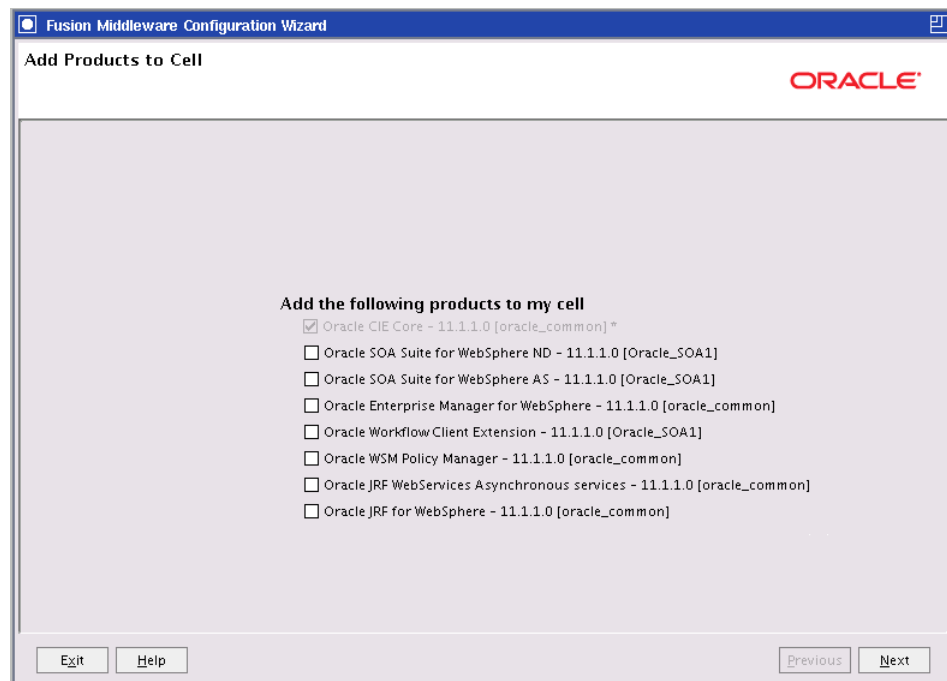


Select the products that you want to include in this cell. By default, the Oracle CIE Core option is selected and cannot be deselected. This option adds the required CIE Online Custom Service to the cell and targets it to OracleAdminServer.

If you are editing an existing cell or federating a machine to a cell, products that have already been included in the cell are already selected and cannot be deselected.

Select the check boxes that correspond to the other products that you want to include in the cell. If you select a product that is dependent on resources provided by another product or products in the list, those products are also selected automatically. Similarly, if you deselect the check box for a product on which another product is dependent, the check box for the dependent product is also deselected.

The list of products (configuration templates) differs based on the Oracle Fusion Middleware product installation that you are configuring.

For more information about Oracle Fusion Middleware product templates, see "Oracle Fusion Middleware Product Templates" in the *Oracle Fusion Middleware Domain Template Reference*. Note that only some of the Oracle Fusion Middleware product templates are supported in an IBM WebSphere installation.

If you are adding products to an existing cell, all products that were previously added to the cell are already selected and cannot be deselected.

After selecting all products to include in the cell, click **Next** to continue.

---

**Note:**   When you click **Next**, if any of the selected products require resources provided by another product that is not selected, a warning dialog box is displayed. Return to the Add Products to Cell screen to select the dependency product before proceeding.

---

## 3.8  Configure the Application Server
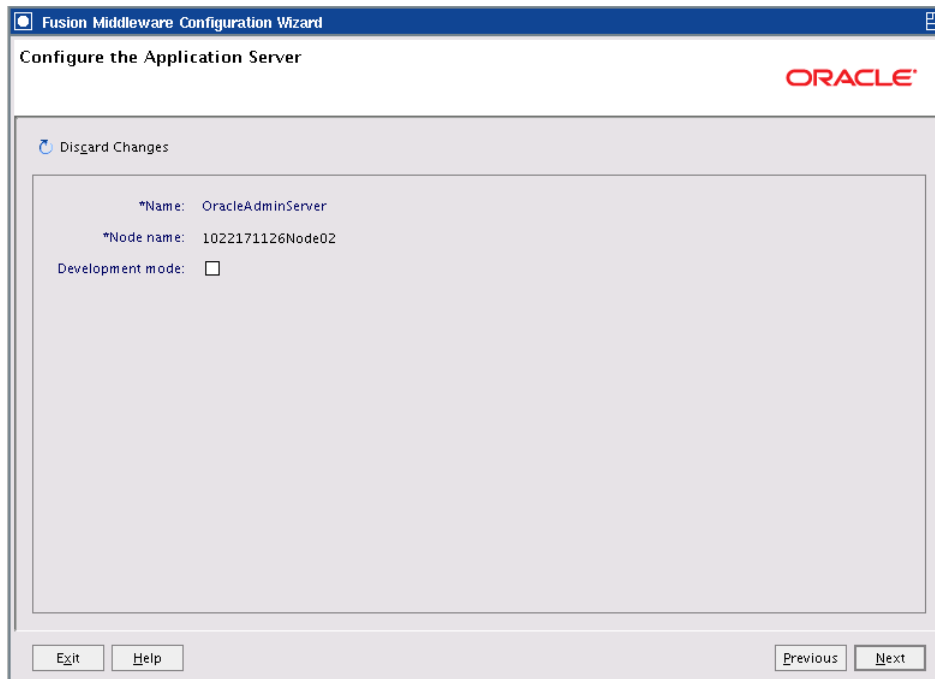


This screen appears only for IBM WebSphere AS. Specify the following Application Server settings:

■  **Name**: The name of the Application Server. It defaults to OracleAdminServer or to the Application Server name that you specified on the Specify Cell, Profile and Node Name Information screen. You cannot change the name.

■  **Node name**: The name of the Application Server node.

■  **Development mode**: Select this check box to run the Application Server in development mode. Leave this check box empty to run the Application Server in production mode. For information about the differences between development mode and production mode, see the IBM WebSphere documentation.

When you are done, click **Next** to continue.

## 3.9 Configure JDBC Data Sources



A JDBC data source contains a pool of database connections that are created when the data source instance is created, when it is deployed or targeted, or at server startup. Applications look up a data source on the JNDI tree, and then request a connection. When the applications no longer need the connections, they return the connections to the connection pool in the data source.

Use this screen to configure the JDBC data sources defined in your cell.

The JDBC data sources associated with the products for which you are creating the cell are listed in the lower half of the screen.

> **Note:** To configure one or more of the data sources as Oracle RAC multi data sources, select the check box adjacent to the name of the data source, and select the **Configure selected data sources as RAC multi data sources in the next panel** check box; then click the **Next** button.

Select the data sources for which you want to specify settings by selecting the check box adjacent to each data source name.

> **Notes:** When you select multiple data sources, the text `Varies among data sources` might be displayed in certain fields, indicating that the current values of those fields are different across the selected data sources. If you change the values in such fields, the new values are applied uniformly across the selected data sources.
>
> The default values of data source parameters such as vendor, driver, host name, and port number depend on the values that are specified in the application templates.

After selecting the data sources, configure the following fields. The settings apply to all selected data sources:

- **Vendor**: Select the database vendor.

- **Driver**: Select the JDBC driver to use to connect to the database. The list includes common JDBC drivers for the selected database vendor.

- **Username**: Enter the user name for connecting to the database.

- **Password**: Enter the password for the specified user name.

- **DBMS/Service**: Enter a DBMS SID or service name. The value that you enter depends on the driver that you selected.

  - If the name of the Oracle driver that you selected contains the words "for Instance connections," you must enter the SID.

  - If the name of the Oracle driver contains the words "for Service connections," you must enter the service name.

  - For information about configuring a DataDirect driver, see the DataDirect documentation.

- **Host Name**: Enter the name of the server hosting the database.

- **Port**: Enter the port number to be used to connect to the server.

For the selected data sources, the values that you specify are displayed in the appropriate columns in the data source list.

When you are finished configuring the JDBC data sources, click **Next** to continue.

## 3.10  Configure RAC Data Sources



Use this screen to configure the data sources that you chose to configure as Oracle RAC data sources.

The data sources that you opted to configure as Oracle RAC data sources on the Configure JDBC Data Sources screen are listed in the lower half of the screen.

Select the data sources for which you want to specify settings by selecting the check box adjacent to each data source name.

> **Note:** When you select multiple data sources, the text `Varies among data sources` might be displayed in certain fields, indicating that the current values of those fields are different across the selected data sources. If you change the values in such fields, the new values are applied uniformly across the selected data sources.

After selecting the data sources, configure the following fields. The settings apply to all selected data sources:

- **Driver**: Select the JDBC driver to use to connect to the database.

- **Service Name**: Enter an Oracle RAC database service name.

- **Username**: Enter the user name for connecting to the database.

- **Password**: Enter the password for the specified user name.

- **Host Name**: Enter the name of the server hosting the Oracle RAC database instances.

- **Port**: Enter the port numbers to be used to connect to the server that hosts the database.

To add a new database instance, click **Add**, and then specify the host name, instance name and port number.

When you are finished configuring your Oracle RAC multi data sources, click **Next**.

## 3.11 Test JDBC Data Sources
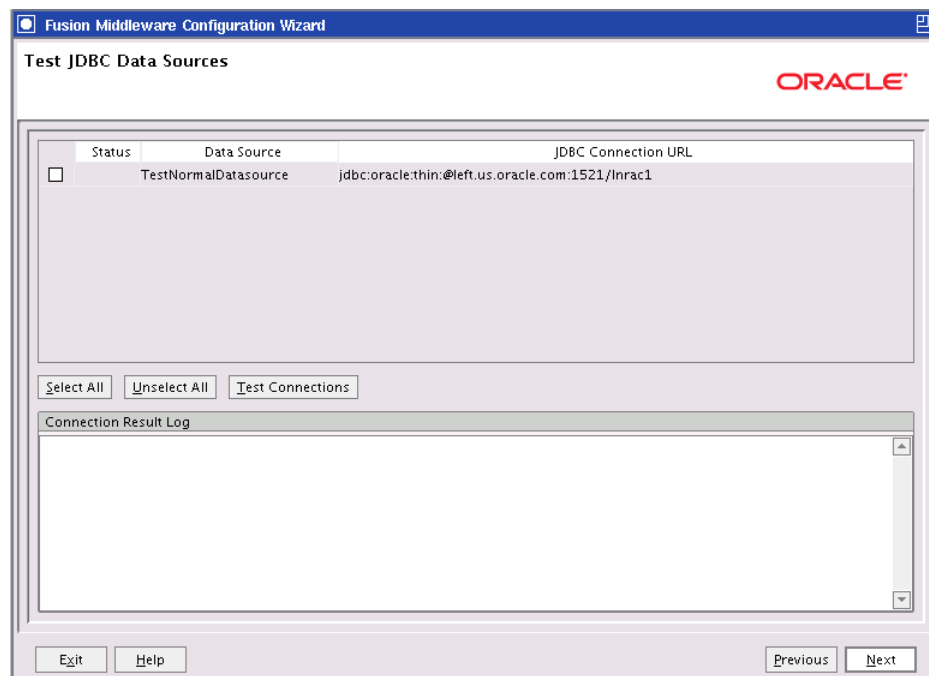
Use this screen to test the data source connections that you configured on the Configure JDBC Data Sources and Configure RAC Data Sources screens.

---

**Notes:**  To test the database connections, the database to which you are connecting must be running.

If you do not want to test the connections at this time, do not select any data sources. Click **Next** to continue, then click **OK** on the warning dialog box that appears.

---

Select the check box for each data source you want to test, and then click **Test Connections**.

The wizard tests the configuration for each selected data source by attempting to connect to a URL that is constructed by using the driver, host, port, and other information that you specified while configuring the data source.

The result of the test is indicated in the **Status** column. Details are displayed in the **Connection Result Log** section.

When the tests complete successfully, click **Next** to continue.

## 3.12  Configure JDBC Component Schema



For some Oracle Fusion Middleware components (for example, Oracle SOA Suite and Oracle Web Services), JDBC data sources might be defined as part of the component's database schema. The schema is loaded (during installation) by using the Oracle Fusion Middleware Repository Creation Utility (RCU).

> **Note:** For a complete list of the schemas that are available for various products for supported database types, see "Required Schemas and Dependencies" in *Oracle Fusion Middleware Repository Creation Utility User's Guide*.

When you create a cell for such components by using the Configuration Wizard, you can configure the JDBC component schema settings, such as database driver, schema owner, password, and so on.

The JDBC component schemas associated with the products for which you are creating the cell (for example, Oracle SOA Suite) are listed in the lower half of the screen.

> **Note:** To configure one or more of the schemas as Oracle RAC multi data sources, select the check boxes adjacent to the schema names, and select the **Configure selected component schemas as RAC multi data source schemas in the next panel** check box. Click **Next** when you are done.

Select the schemas for which you want to specify data source settings by selecting the check box adjacent to each schema name.

> **Note:** When you select multiple component schemas, the text `Varies among component schemas` might be displayed in certain fields, indicating that the current values of those fields are different across the selected component schemas. If you change the values in such fields, the new values are applied uniformly across the selected component schemas.
>
> The default values of component schema parameters such as vendor, driver, host name, and port number depend on the values that are specified in the application templates.

After selecting the schemas, configure the following fields. The settings apply to all selected schemas:

- **Vendor**: Select the database vendor.

- **Driver**: Select the JDBC driver to use to connect to the database. The list includes common JDBC drivers for the selected database vendor.

- **Schema Owner**: Enter the user name for connecting to the database.

- **Schema Password**: Enter the password for the specified schema owner.

- **DBMS/Service**: Enter a database DBMS name, or service name if you selected a service type driver.

- **Host Name**: Enter the name of the server hosting the database.

- **Port**: Enter the port number to be used to connect to the server that hosts the database.

For the selected schemas, the values that you specify are displayed in the appropriate columns in the schema list.

When you finish configuring the component schemas, click **Next** to continue.

## 3.13  Configure RAC Data Source Component Schema



Use this screen to configure the component schemas included in the cell as Oracle RAC schemas.

The component schemas that you selected on the Configure JDBC Component Schema screen for configuration as Oracle RAC schemas are listed on the lower half of the screen.

> **Note:**   For a complete list of the schemas that are available for various products for supported database types, see "Required Schemas and Dependencies" in *Oracle Fusion Middleware Repository Creation Utility User's Guide*.

In the schema list in the lower half of the screen, select the schemas to configure as Oracle RAC schemas by selecting the check box adjacent to each schema name.

> **Note:**   When you select multiple schemas, the text "`Varies among component schemas`" might be displayed in certain fields, indicating that the current values of those fields are different across the selected schemas. If you change the values in such fields, the new values are applied uniformly across the selected schemas.

After selecting the schemas, configure the following fields. The settings apply to all selected schemas:

- **Driver**: Select the JDBC driver to use to connect to the database.

- **Service Name**: Enter a database Oracle RAC service name.

- **Username**: Enter the user name for connecting to the database.

- **Password**: Enter the password for the specified user name.

- **Host Name**: Enter the name of the server hosting the Oracle RAC database instances.

- **Port**: Enter the port number to be used to connect to the server that hosts the database.

---

> **Note:** You must specify the host name, instance name, and port number of at least one database instance.
>
> To add a database instance, click **Add**. Specify the host name, instance name, and port.

---

For the selected schemas, the values that you specify are displayed in the appropriate columns in the schema list.

## 3.14 Test Component Schema
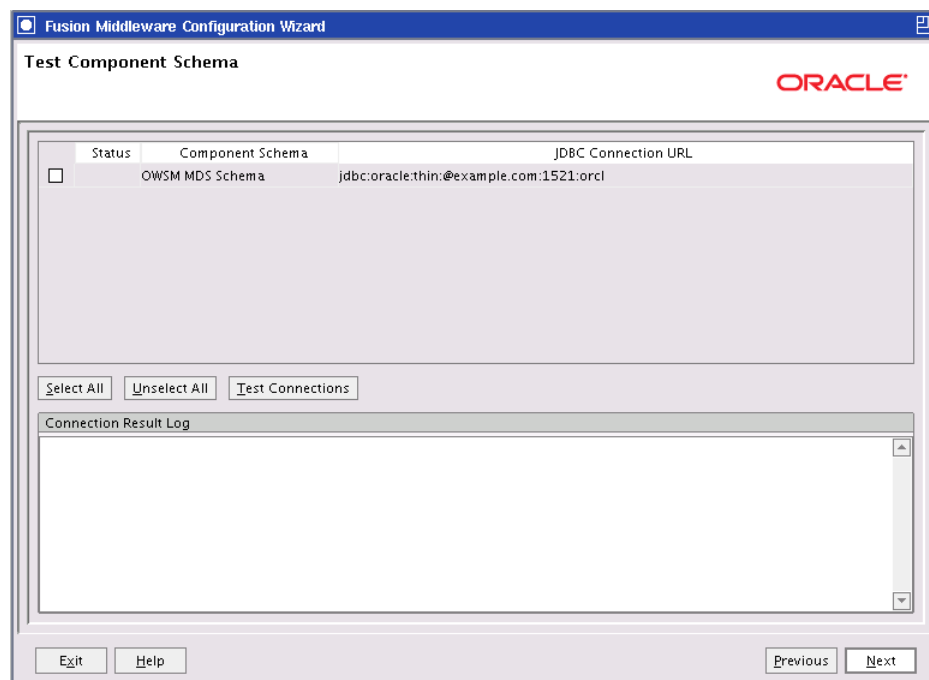


Use this screen to test the configurations that you specified for the component schemas on the previous screen.

Select the check boxes adjacent to the names of the schemas to test, and then click **Test Connections**.

The wizard tests the configuration for each schema by attempting to connect to a URL that is constructed by using the driver, host, port, and other information that you specified while configuring the schema.

The result of the test is indicated in the **Status** column. Details are displayed in the **Connection Result Log** section.

When the schema tests complete successfully, click **Next** to continue.

## 3.15 Select Optional Configuration
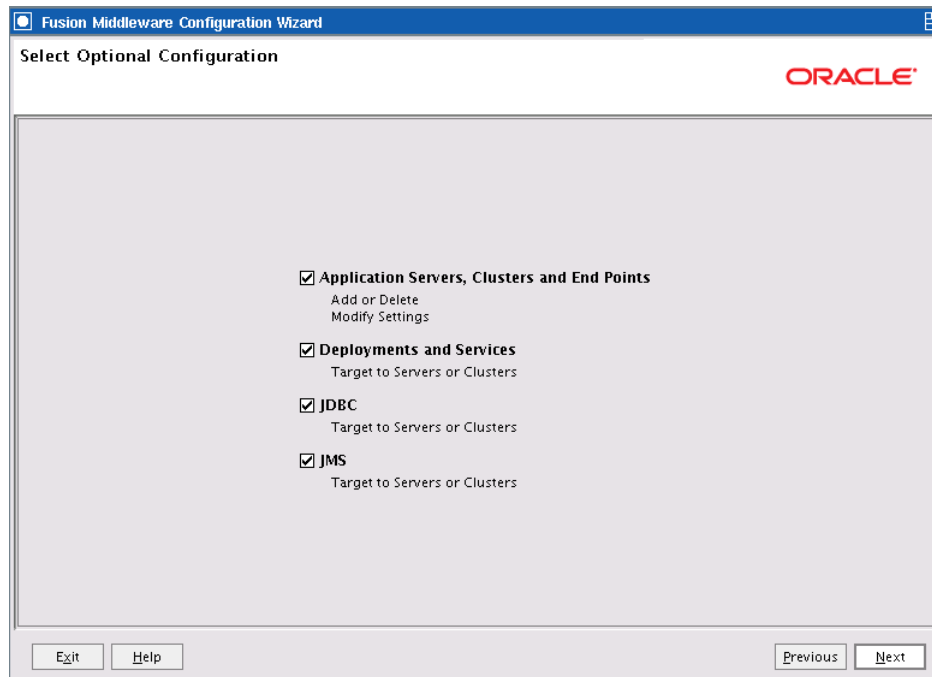


Select the categories (if any) for which you want to perform advanced configuration. For each category that you select, the appropriate configuration screen is displayed to allow you to perform advanced configuration. If you do not select any items on this screen, the Configuration Summary screen is displayed next.

> **Notes:** The categories that are listed on this screen depend on the resources defined in the templates that you selected for the cell.

■ **Application Servers, Clusters, and End Points**: This option is available only in an IBM WebSphere ND environment. It is not available in an IBM WebSphere AS environment.

Select this option to perform any of the following tasks:

– Add, delete, or clone an Application Server

– Change the name of an Application Server

– Toggle development mode on or off for an Application Server

– Add clusters to the cell, and add servers to a cluster

– Configure end points (ports) for the Administration Server or any Application Server

– Change the file store shared directory locations for your clusters, including the log directory, permanent store directory, and temporary store directory

■ **Deployments and Services**: Select this option to perform any of the following tasks. In an IBM WebSphere AS environment, you can only enable or disable the items that are targeted to the Application Server.

– Change the deployment targets for your clusters or servers

– Change the service targets for your clusters or servers

- – Change the custom service targets for your servers

- – Change the resource adapter targets for your servers

- **JDBC**: Select this option to change the JDBC data source targets for your clusters or servers. In an IBM WebSphere AS environment, you can only enable or disable the data sources that are targeted to the Application Server.

- **JMS**: Select this option to change the JMS resource targets for your clusters or servers. In an IBM WebSphere AS environment, you can only enable or disable the JMS resources that are targeted to the Application Server.

After selecting the appropriate configuration options, click **Next** to continue. The screen that is displayed next differs depending on the options that you selected.

## 3.16 Configure Application Servers
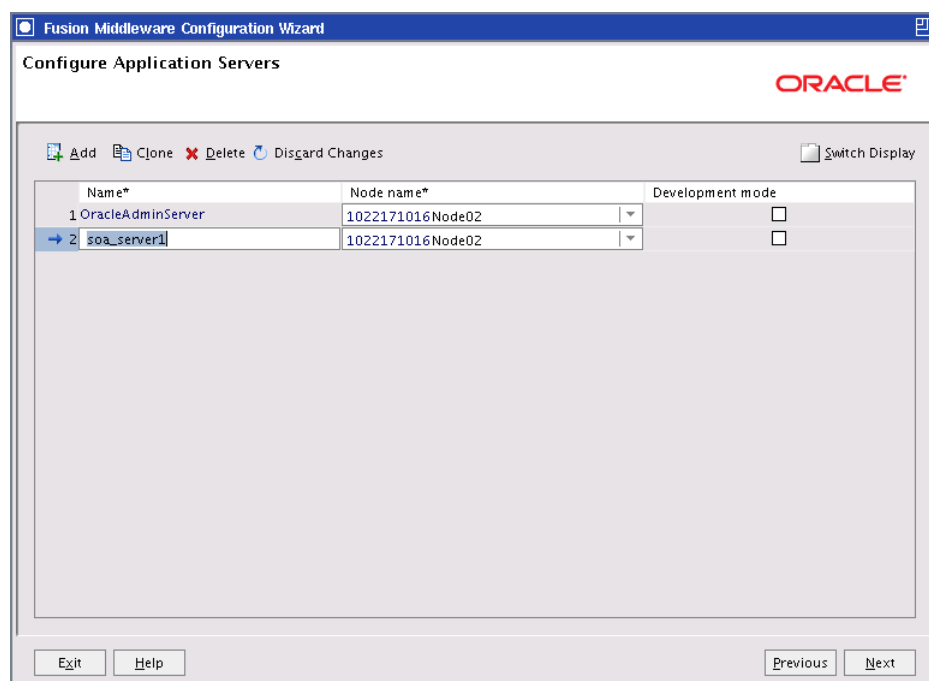


This screen is displayed only in an IBM WebSphere ND environment. From this screen, you can add, clone, or delete Application Servers to any node in the cell. You can also change the settings for an existing Application Server. Required fields are indicated by an asterisk (*).

> **Note:** Use this screen only to create Application Servers that will either (a) be the primary member of a cluster or (b) will not be part of a cluster. Do not create secondary cluster members on this screen. Instead, use the Configure Additional Cluster Members screen to add secondary cluster members to a cluster.
>
> For example, suppose you want to create a three-server cluster with servers soa_server_1, soa_server_2, and soa_server_3. To do so, add the soa_server_1 Application Server on the Configure Application Servers screen. Create the cluster on the Configure Clusters screens, and select soa_server_1 as the primary cluster member. You can then use the Configure Additional Cluster Members screen to add soa_server_2 and soa_server_3 to the cluster.

- **Name**: The name of the Application Server. The default name for a new server is new_ManagedServer_*n*, where *n* starts at 1 and increments for each new server that you add. To change the server name, click in the **Name** field and enter the new name.

- **Node name**: A list of all available nodes. Click the list to select the node on which this Application Server resides. If you have not yet federated other machines to the current cell, only one node is available on this list.

- **Development mode**: Select this check box to run the Application Server in development mode. Leave this check box empty to run the Application Server in production mode. For information about the differences between production mode and development mode in an IBM WebSphere environment, see the IBM WebSphere documentation.

To add a server, click **Add** and configure the settings for the new server.

To clone an existing server, select the server that you want to clone and click **Clone**. A new server line is added to the table. The new server inherits all attributes from the existing server. The default name of the new server is *clone_source*_cloned1, where *clone_source* is the name of the server that you cloned. Edit the **Name** field, select the node on which the server resides, and select or deselect the **Development mode** option.

To delete a server, select the server and click **Delete**. You can delete only one server at a time.

To switch the display from the default table format to a tabbed format (with one tab for each server), click **Switch Display**. Click **Switch Display** again to return to table format.

When you are done adding servers, click **Next** to continue.

> **Note:** Any changes that you make on this screen are not saved in the cell configuration until you click **Create** on the last Configuration Wizard screen. If you exit the wizard, or click **Previous** to return to an earlier screen, you must make the changes again.

## 3.17 Configure Clusters
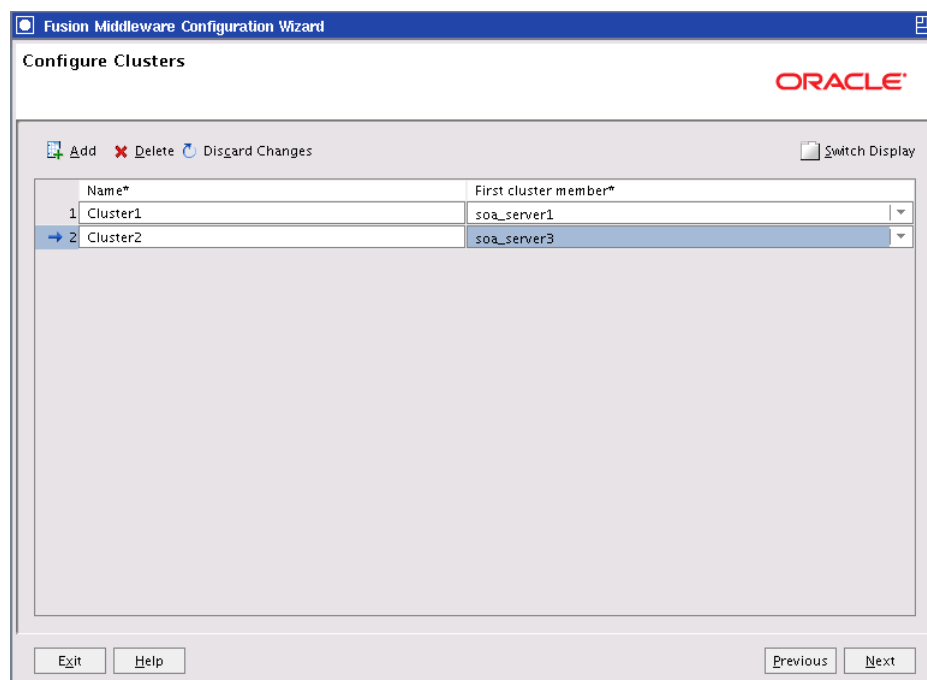


A cluster is a group of Application Servers that work together to provide scalability and high availability for applications. By creating clusters, you can group servers such that they operate as a single unit for hosting applications and resources.

Use this screen to add a cluster and select the Application Server that will be the primary cluster member of the cluster. All additional cluster members are cloned from this primary cluster member. After creating a cluster, you can add additional Application Servers to the cluster on the next screen.

You can also use this screen to delete a cluster or modify the settings for an existing cluster.

To create a cluster:

1. Click **Add**. A new line appears in the table. The default name for a new cluster is new_Cluster_*n*, where *n* starts at 1 and increments for each new cluster that you add. To change the default, enter the new name in the **Name** column. The name of the cluster must be unique among all component names within the cell.

2. Select an existing Application Server as the primary member of the cluster by selecting the server from the **First cluster member** list. Only Application Servers that are not yet assigned to a cluster are displayed on this list.

3. Repeat Steps 1 and 2 for each additional cluster that you want to create.

4. When you are done, click **Next**. You can add additional servers to the cluster on the Configure Additional Cluster Members screen, which is displayed when you click **Next**.

> **Note:** Any changes you make on this screen are not saved in the cell configuration until you click **Create** on the last Configuration Wizard screen. If you exit the wizard, or click **Previous** to return to an earlier screen, you must make the changes again.

To delete a cluster, select the cluster and click **Delete**. When you delete a cluster, you do not delete the Application Server that is the primary cluster member; it is merely removed from the cluster. You can delete only one cluster at a time.

To change the settings for a cluster, select the cluster and make the appropriate changes.

To switch the display from the default table format to a tabbed format (with one tab for each cluster), click **Switch Display**. Click **Switch Display** again to return to table format.

When you are done configuring your clusters, click **Next** to continue.

For more information about clusters in an IBM environment, see the IBM WebSphere documentation.

# 3.18  Configure Additional Cluster Members
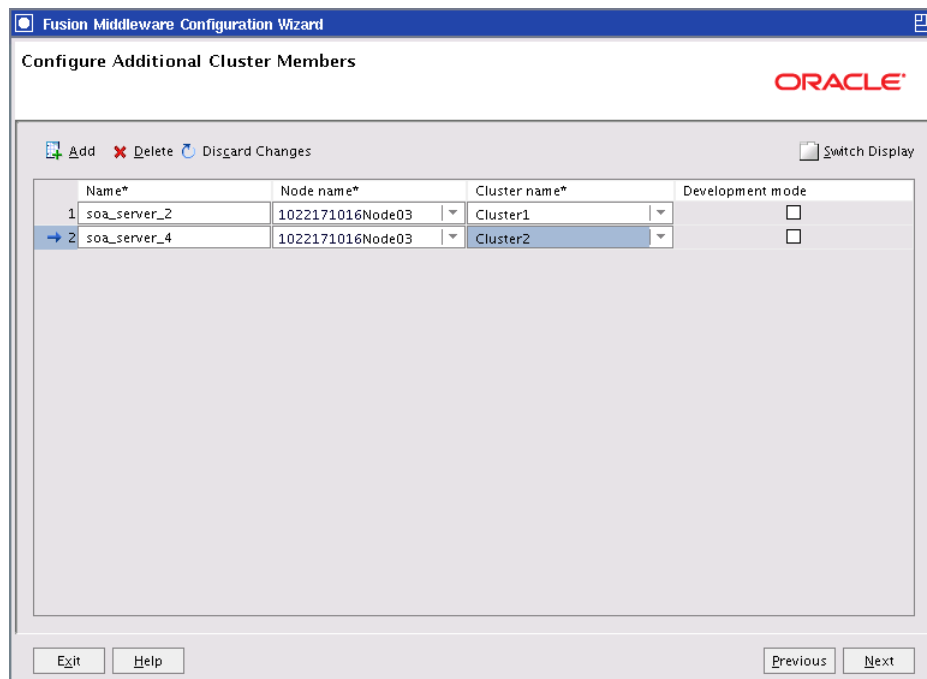


Use this screen to add additional Application Servers to a cluster. When you add a new Application Server to a cluster, its attributes are cloned from the primary member of the cluster.

Configure the following settings for the Application Server:

- **Name**: The name of the Application Server. The default name for a new server is new_ClusterMember_$n$, where $n$ starts at 1 and increments for each new server you add. To change the server name, click in the **Name** field and enter the new name.

- **Node name**: A list of all available nodes. Click the list to select the node on which this Application Server resides. If you have not yet federated other machines to the current cell, only one node is available on this list.

- **Cluster name**: A list of all available clusters. Select the cluster to which you want to assign this server.

- **Development mode**: Select this check box to run the Application Server in development mode. Leave this check box empty to run the Application Server in production mode. For information about the differences between production mode and development mode in an IBM WebSphere environment, see the IBM WebSphere documentation.

To add an additional cluster member:

1. Click **Add**. A new row is added to the table.

2. Enter the name of the server, and select the appropriate node, cluster, and development mode setting.

3. Repeat Steps 1 and 2 for each cluster member that you want to add.

To delete a cluster member, select the cluster member and click **Delete**. You can delete only one cluster member at a time.

To switch the display from the default table format to a tabbed format (with one tab for each cluster), click **Switch Display**. Click **Switch Display** again to return to table format.

When you are done, click **Next** to continue.

---

**Note:** Any changes that you make on this screen are not saved in the cell configuration until you click **Create** on the last Configuration Wizard screen. If you exit the wizard, or click **Previous** to return to an earlier screen, you must make the changes again.

---

## 3.19 Configure End Points



Use this screen to configure the end points (listener addresses) on each of your servers, including the Administration Server. All end points must be unique in the cell. By default, unique end points are assigned to a server when you add the server to the cell.

You do not have to modify them unless you want to use a port other than the default that was configured on that server.

For information about the various end points listed on this screen, see the IBM WebSphere documentation.

---

**Note:** An asterisk (*) in the **Host** column indicates any local address.

---

To change end points:

1. (IBM WebSphere ND only) From the **Server Name / Node Name** list, select the server for which you want to change end points. The list contains all servers in the cell, including the Administration Server, Application Servers that are assigned to clusters, and standalone Application Servers that exist on all nodes in the cell.

   ---

   **Note:** In an IBM WebSphere AS environment, there is only one server. The name of this server and node is displayed in the **Server Name / Node Name** field, and cannot be changed.

   ---

2. (IBM WebSphere ND only) You can change the host name on which a particular end point resides. To change the host name for an end point, enter the new host name on the appropriate row of the **Host** column.

   To use a common host for all end points on the selected server, select the **Use common host for all end points** check box. When this check box is selected, the host names for each end point on the currently selected server cannot be edited, except for the **Admin end point** host name.

3. In the **End Point Type** column, enter the new end point value on the appropriate row. If you enter a value that is already in use, the following validation error is displayed; click **OK** and enter another value for the end point.



   ---
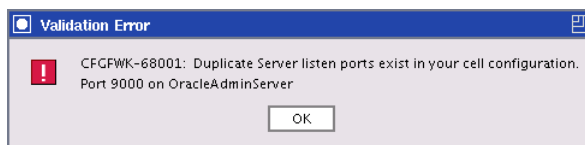
   **Notes:** If you plan to run multiple Deployment Managers on the same machine, you must ensure that the end points in each associated cell are unique across all Deployment Manager nodes on the machine.

   Similarly, if you have Application Servers on remote machines, and those servers are assigned to different cells on the Deployment Manager host, you must ensure that all end points for all Application Servers on a machine are unique.

   You will not be warned about conflicting end points that exist in different cells.

   ---

4. Repeat Step 3 for each end point that you want to change.

5. You can also disable an end point on any server if you do not need it. To disable one or more end points on the selected server, deselect the check box on the

appropriate row of the **Enabled** column for each end point that you want to disable.

6. When you are done, click **Next**.

If there are any end point conflicts, the Validation Error dialog box appears, and you must resolve the conflict before you can continue. Identify the conflicting ports, and change the appropriate one.

## 3.20 Configure File Store Shared Directories
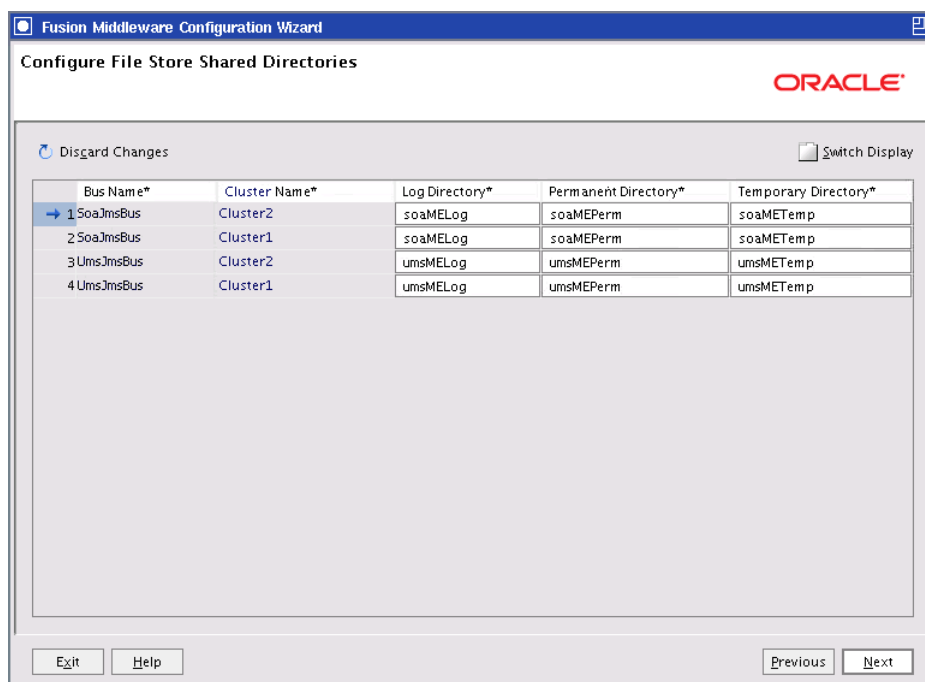


Use this screen to change the file store shared directories for each of the JMS service integration buses in the cell. For more information about JMS service integration buses, see the IBM WebSphere documentation.

The table contains the following columns.

- **Bus Name**: The name of the service bus as defined in the product template's `config.xml` file. A service bus can be used by multiple clusters, and therefore may have multiple rows in the table.

- **Cluster Name**: The name of each cluster that uses a service bus. A cluster uses a bus if the product to which the bus applies is targeted to that cluster.

- **Log Directory**: The directory in which the log files for the bus will be stored. You can change the directory location.

- **Permanent Directory**: The directory in which the permanent file store for the bus will be stored. You can change the directory location.

- **Temporary Directory**: The directory in which the temporary file store for the bus will be stored. You can change the directory location.

> **Note:** Oracle recommends that you change the default directory locations. To change a directory location, enter an *absolute* path to the new directory location that you want to use. The directories are created when you start one of the Application Servers on the bus.

If you keep the default directory locations, these directories are created under the Application Server profile directory for the cell. For example, if the Application Server profile name for the cell is Custom01 on a UNIX operating system, the directories are created under *WAS_HOME*/profiles/Custom01.

When you are done, click **Next**.

## 3.21 Target Deployments to Clusters or Servers
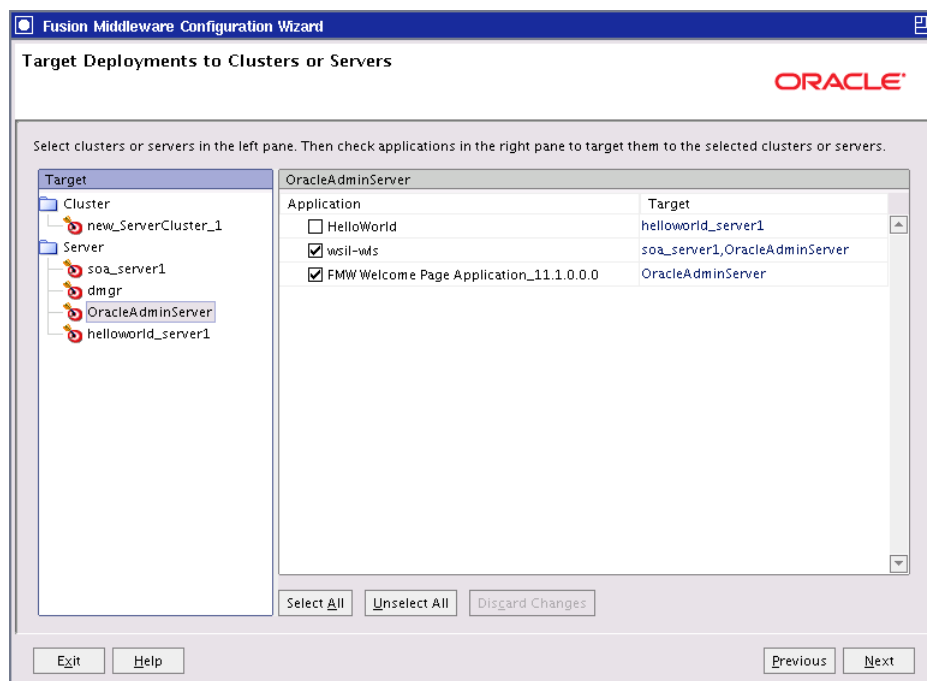


Use this screen to target applications for deployment on servers or clusters.

Applications associated with the product for which you are configuring the cell (for example, Oracle SOA Suite) are targeted automatically to the server that was created for that product or to the cluster to which that server is assigned. In this screen, you can target applications to additional servers and clusters.

To target an application deployment to a cluster or server:

1.  In the **Target** list, select the cluster or server on which you want to deploy applications.

    The name of the selected target is displayed as the title of the list on the right.

2.  In the *target_name* list, select the check boxes corresponding to the applications to deploy on the selected target.

    The applications displayed here vary, depending on the products that you selected in the Add Products to Cell screen, earlier in the wizard.

> **Note:** When you select a server in the **Target** list, some of the check boxes in the *target_name* list might be disabled, indicating applications that are already targeted at the cluster that contains the selected server.

After you select applications, the names of the targeted clusters and servers are displayed in the Target column in the *target_name* list.

3.  Repeat Steps 1 and 2 for the other clusters and servers, as required.

4.  After making the required selections, click **Next**.

When you are updating an existing cell, if you delete a server or cluster to which applications are currently targeted, the Configuration Wizard automatically retargets those applications as follows:

■   If the applications were originally targeted solely to the server or cluster that you are now deleting (that is, after you delete the server or cluster, the applications would become untargeted in the modified cell), then the Configuration Wizard automatically retargets the applications to all *eligible* targets.

An eligible target is any cluster or server that is not defined in the configuration groups specification (`config-groups.xml` file) of an included template. Servers or clusters that are specified in `config-groups.xml` are essentially owned by the template and, therefore, are not eligible for automatic targeting.

■   If the applications were originally targeted to multiple targets (including servers, clusters, and the Administration Server), and one of the targeted servers or clusters is deleted, then the Configuration Wizard leaves the remaining target associations intact and does not attempt to retarget the applications.

## 3.22  Target Services to Clusters or Servers
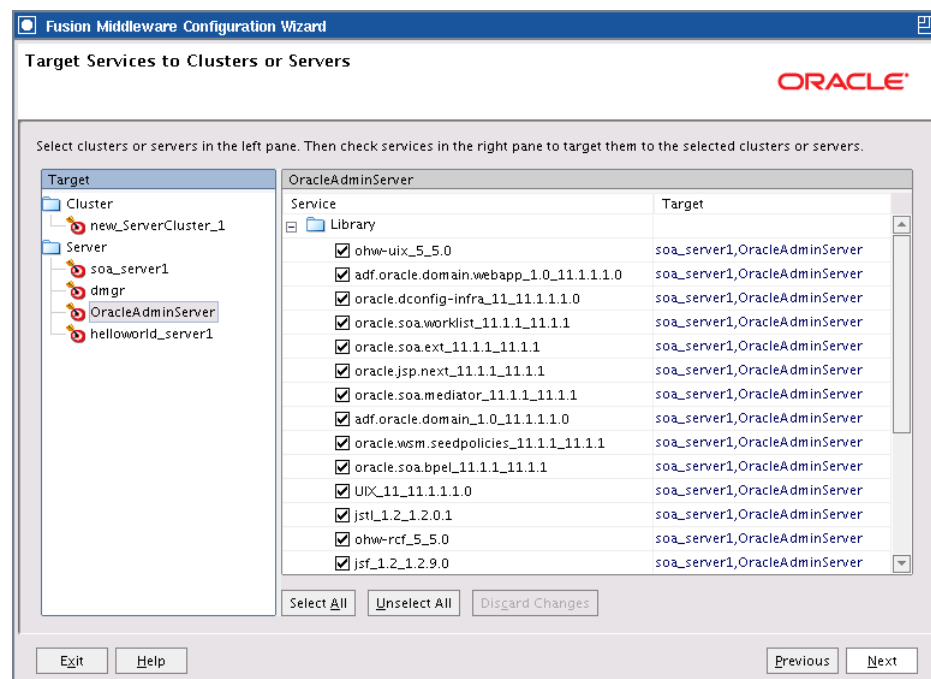
Use this screen to target services (such as libraries, JMS services, or Work Managers) to the appropriate servers or clusters.

Services that are associated with the product for which you are configuring the cell (for example, Oracle SOA Suite) are targeted automatically to the server that was created for that product or to the cluster to which that server is assigned. In this screen, you can target services to additional servers and clusters.

To target services to servers or clusters:

1. In the **Target** list, select the cluster or server on which you want to deploy services.

   The name of the selected target is displayed as the title of the list on the right.

2. In the *target_name* list, select the check boxes corresponding to the services to deploy on the selected target.

   The services displayed here vary, depending on the products that you selected in the Add Products to Cell screen earlier in the wizard.

   > **Note:** When you select a server in the **Target** list, some of the check boxes in the *target_name* list might be disabled, indicating services that are already targeted at the cluster that contains the selected server.

   After you select services, the names of the targeted clusters and servers are displayed in the **Target** column in the *target_name* list.

3. Repeat Steps 1 and 2 for the other clusters and servers, as required.

4. After making the required selections, click **Next**.

When you are updating an existing cell, if you delete a server or cluster to which services are currently targeted, the Configuration Wizard automatically retargets those services as follows:

- If the services were originally targeted solely to the server or cluster that you are now deleting (that is, after you delete the server or cluster, the services would become untargeted in the modified cell), then the Configuration Wizard automatically retargets the services to all *eligible* targets.

  An eligible target is any cluster or server that is not defined in the configuration groups specification (`config-groups.xml` file) of an included product template. Servers or clusters that are specified in `config-groups.xml` are essentially owned by the template and, therefore, are *not* eligible for automatic targeting.

- If the services were originally targeted to multiple targets (including servers, clusters, and the Administration Server), and one of the targeted servers or clusters is deleted, then the Configuration Wizard leaves the remaining target associations intact and does not attempt to retarget the services.

## 3.23  Target Custom Services to Servers
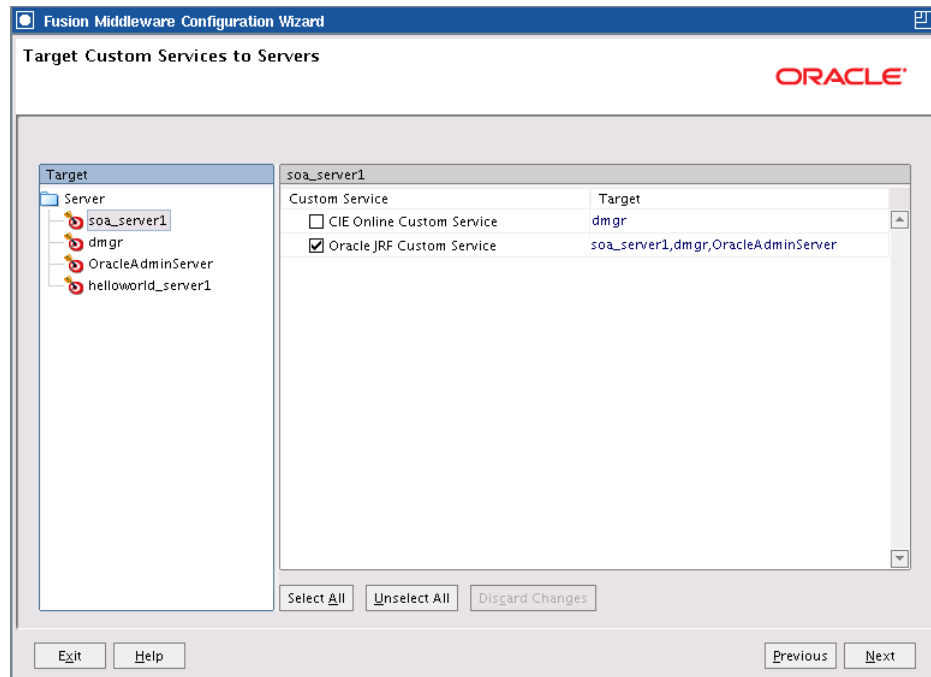


Use this screen to target custom services to the appropriate servers. You cannot target custom services to clusters. Servers that are assigned to clusters are included in the **Target** list.

Custom services that are associated with the product for which you are configuring the cell (for example, Oracle SOA Suite) are targeted automatically to the server that was created for that product. In this screen, you can target custom services to additional servers.

To target custom services to servers:

1.  In the **Target** list, select the server to which you want to deploy services.

    The name of the selected target is displayed as the title of the list on the right.

2.  In the *target_name* list, select the check boxes corresponding to the services you want to deploy to the currently selected target.

    The services displayed here vary, depending on the products that you selected in the Add Products to Cell screen earlier in the wizard.

    The **Target** column in the *target_name* list displays the names of *all* servers to which each custom service is currently targeted. This helps you to determine if you need to target a custom service to any other servers other than the currently selected target.

3.  Repeat Steps 1 and 2 for the other servers, as required.

4.  After making the required selections, click **Next**.

When you are updating an existing cell, if you delete a server to which custom services are currently targeted, the Configuration Wizard automatically retargets those services as follows:

■   If the custom services were originally targeted solely to the server that you are now deleting (that is, after you delete the server, the custom services would

become untargeted in the modified cell), then the Configuration Wizard automatically retargets the custom services to all *eligible* targets.

An eligible target is any server that is not defined in the configuration groups specification (`config-groups.xml` file) of an included product template. Servers that are specified in `config-groups.xml` are essentially owned by the template and, therefore, are *not* eligible for automatic targeting.

- If the custom services were originally targeted to multiple targets (including servers and the Administration Server), and one of the targeted servers is deleted, then the Configuration Wizard leaves the remaining target associations intact and does not attempt to retarget the services.

## 3.24 Target Resource Adapters to Servers
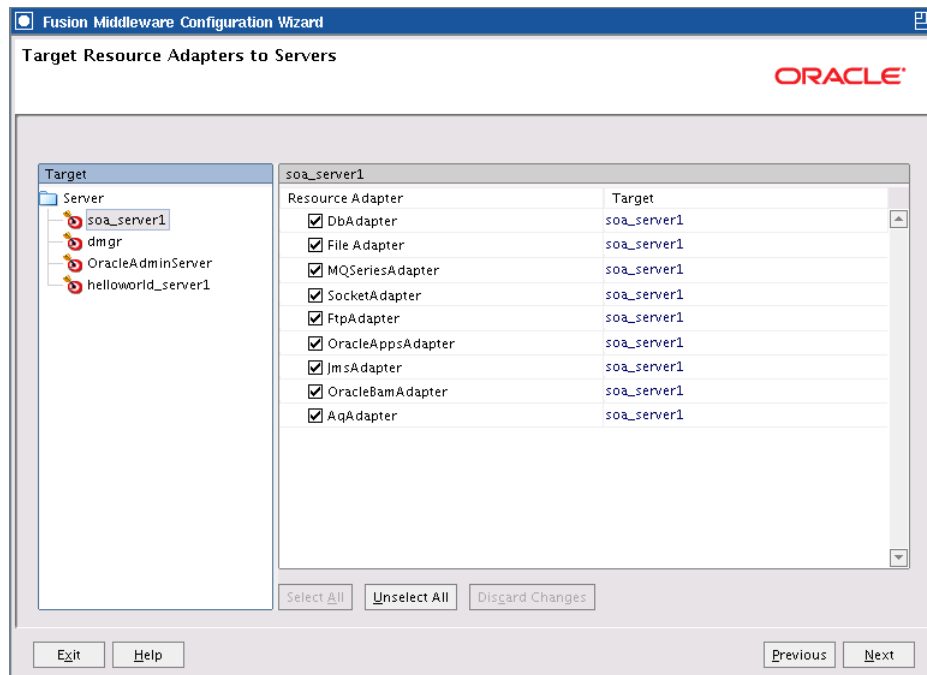


Use this screen to target resource adapters to individual servers. You cannot target resource adapters to clusters. Servers that are assigned to clusters are included in the **Target** list.

Resource adapters that are associated with the product for which you are configuring the cell (for example, Oracle SOA Suite) are targeted automatically to the server that was created for that product. In this screen, you can target resource adapters to additional servers.

To target resource adapters to servers:

1. In the **Target** list, select the server on which you want to target resource adapters.

   The name of the selected target is displayed as the title of the list on the right.

2. In the *target_name* list, select the check boxes corresponding to the resource adapters that you want to target to the selected server.

   The resource adapters that are displayed here vary, depending on the products that you selected in the Add Products to Cell screen earlier in the wizard.

After you select resource adapters, the names of the targeted servers are displayed in the **Target** column in the *target_name* list.

3. Repeat Steps 1 and 2 for the other servers, as required.

4. After making the required selections, click **Next**.

When you are updating an existing cell, if you delete a server to which resource adapters are currently targeted, the Configuration Wizard automatically retargets those resource adapters as follows:

■ If the resource adapters were originally targeted solely to the server that you are now deleting (that is, after you delete the server, the resource adapters would become untargeted in the modified cell), then the Configuration Wizard automatically retargets the resource adapters to all *eligible* targets.

An eligible target is any server that is not defined in the configuration groups specification (`config-groups.xml` file) of an included product template. Servers that are specified in `config-groups.xml` are essentially owned by the template and, therefore, are *not* eligible for automatic targeting.

■ If the resource adapters were originally targeted to multiple targets (including servers and the Administration Server), and one of the targeted servers is deleted, then the Configuration Wizard leaves the remaining target associations intact and does not attempt to retarget the resource adapters.

## 3.25 Target JDBC Data Sources to Clusters or Servers
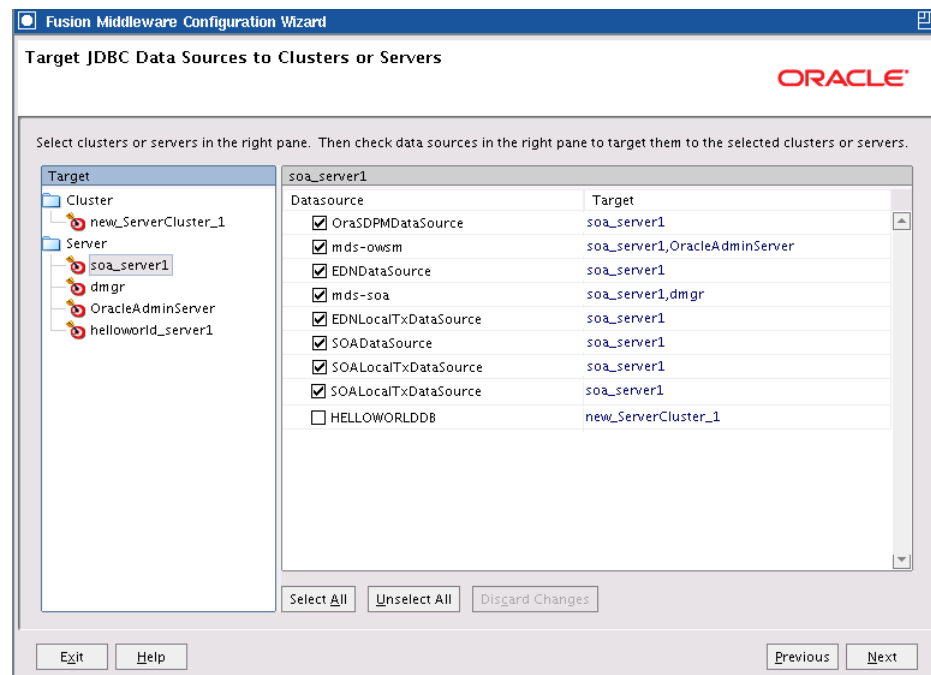


Use this screen to target JDBC data sources to servers or clusters.

To target a JDBC data source to a cluster or server:

1. In the **Target** list, select the cluster or server to which you want to target a data source.

   The name of the selected target is displayed as the title of the list on the right.

**2.** In the *target_name* list, select the check boxes corresponding to the data sources that you want to target to the selected server or cluster.

> **Note:** When you select a server in the **Target** list, some of the check boxes in the *target_name* list might be disabled, indicating data sources that are already targeted at the cluster that contains the selected server.

After you select data sources, the name of the targeted cluster or server is displayed in the **Target** column in the *target_name* list.

**3.** Repeat Steps 1 and 2 for the other clusters and servers, as required.

**4.** After making the required selections, click **Next**.

## 3.26  Target JMS Resources to Clusters or Servers
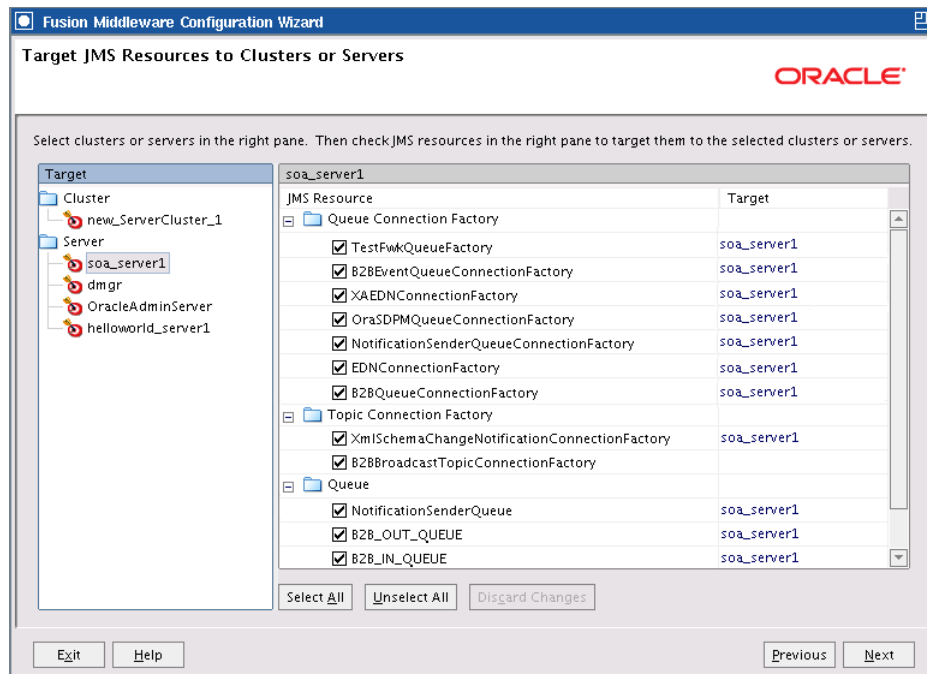


Use this screen to target Java Message Service (JMS) resources to servers or clusters.

To target a JMS resource to a cluster or server:

**1.** In the **Target** list, select the cluster or server to which you want to target the JMS resource.

The name of the selected target is displayed as the title of the list on the right.

**2.** In the *target_name* list, select the check boxes corresponding to the JMS resources that you want to target to the selected server or cluster.

> **Note:** When you select a server in the **Target** list, some of the check boxes in the *target_name* list might be disabled, indicating JMS resources that are already targeted at the cluster that contains the selected server.

After you select JMS resources, the name of the targeted cluster or server is displayed in the **Target** column in the *target_name* list.

3. Repeat Steps 1 and 2 for the other clusters and servers, as required.

4. After making the required selections, click **Next**.

## 3.27 Configuration Summary
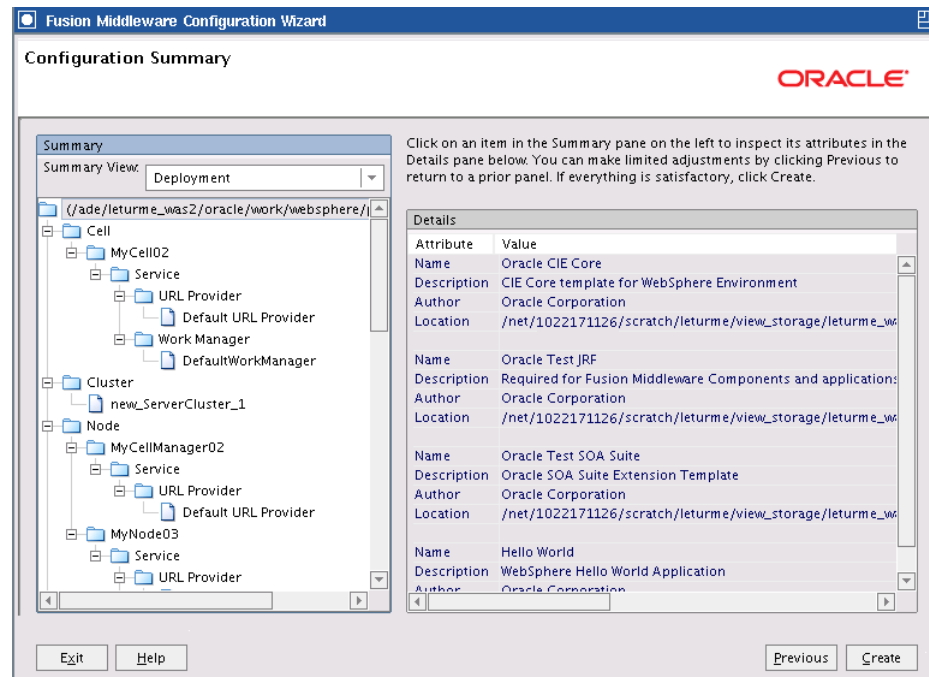


Review the detailed configuration settings of your cell before continuing.

In the **Summary** section on the left, select an item to display details about that item in the **Details** section on the right.

You can limit the items that are displayed in the **Summary** section by selecting a filter option from the **Summary View** list. For example, to view the JDBC data sources that are assigned to each of your clusters and servers, select **JDBC** from the **Summary View** list.

To change the configuration, click **Previous** to return to the appropriate screen.

When you are done reviewing the configuration, click:

- **Create** (if you are creating a new cell)
- **Extend** (if you are modifying an existing cell or federating a machine)

## 3.28  Configuring Cell
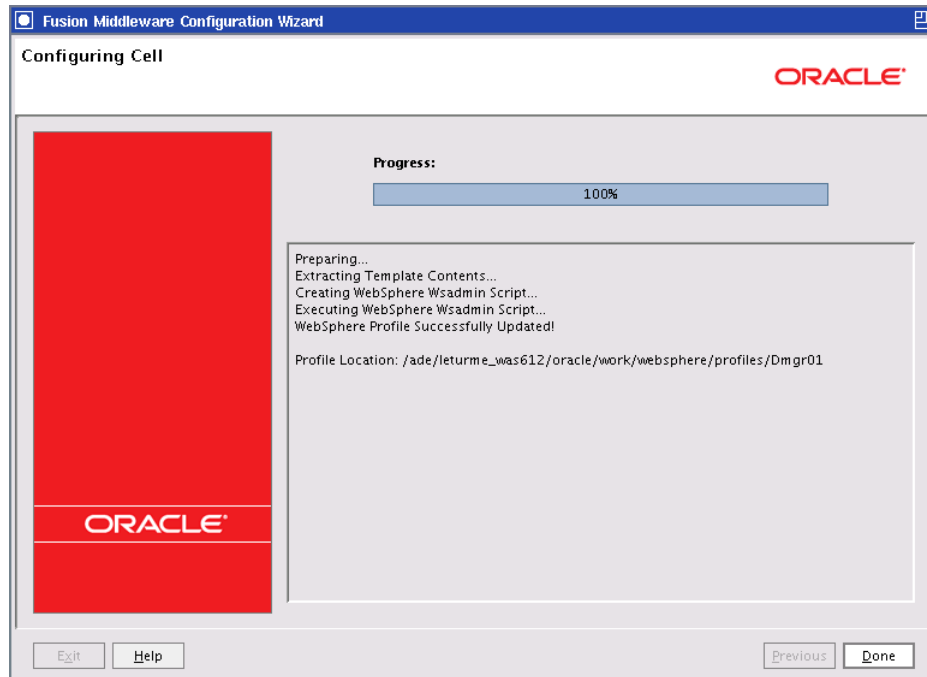


This screen shows the progress of the cell configuration process.

When the cell configuration process completes, click **Done** to close the Configuration Wizard.

On Windows operating systems, to start the Deployment Manager after closing the Configuration Wizard, select the **Start Deployment Manager** check box. This check box is not available on UNIX operating systems.

# 4

# Using wsadmin to Configure Oracle Fusion Middleware

The `wsadmin` tool is a command-line utility that you can use in interactive mode to configure IBM WebSphere cells. You can also use `wsadmin` commands in Jython scripts to automate cell management.

This chapter provides various examples of how to use `wsadmin` commands to perform configuration tasks. It contains the following sections:

- Prerequisite Environment Setup
- Starting and Ending a Configuration Session
- Adding an Oracle Fusion Middleware Product to a Cell
- Adding a New Server to a Cell
- Configuring Host and Port Settings for End Points
- Creating Servers, Clusters, and Cluster Members
- Setting File Store Shared Directories for a Cluster
- Creating and Deleting a Data Source
- Changing a Data Source to an Oracle RAC Data Source
- Changing an Oracle RAC Data Source to a Data Source
- Configuring a Component Schema
- Federating Remote Nodes to a Cell

> **Note:** In the examples in this chapter, any output shown for a `wsadmin` command is displayed only if the command is executed in interactive mode. Output is not displayed when `wsadmin` scripts are executed.

For information about Oracle-specific `wsadmin` commands, see Chapter 5, "wsadmin Command Reference." For additional information about `wsadmin`, see "Using the Oracle Fusion Middleware wsadmin Commands" in *Oracle Fusion Middleware Third-Party Application Server Guide*.

## 4.1 Prerequisite Environment Setup

Before you can use `wsadmin` to configure Oracle Fusion Middleware products in a cell, you must use the IBM WebSphere `manageprofiles.sh` command (UNIX) or

manageprofiles.cmd command (Windows) to set up the initial Deployment Manager profile and the empty custom profile that are needed for configuration. The following sections describe how to do this in an IBM WebSphere ND environment and an IBM WebSphere AS environment.

> **Note:** You do not have to perform the steps in this section if you have already used the Configuration Wizard to create the initial Deployment Manager profile and custom profile.

## 4.1.1 Environment Setup for IBM WebSphere ND

Before you can use wsadmin to configure your Oracle Fusion Middleware products in an IBM WebSphere ND environment, you must first create a management profile with a serverType of DEPLOYMENT_MANAGER. You must also federate (add) an empty custom profile to the Deployment Manager.

The following script uses IBM WebSphere commands to:

- Create the Deployment Manager profile and Deployment Manager on the local machine

- Set the Administrator user name and password for the Deployment Manager

- Create an empty custom profile

- Start the Deployment Manager (required to federate the custom node)

- Federate the custom node to the local machine

- Shut down the custom node

- Shut down the Deployment Manager

> **Note:** The commands used to set up the environment in a Windows operating system are the same as in a UNIX operating system, except that all paths use backslash (\) instead of forward slash (/), and all command names have a .cmd extension instead of a .sh extension.

The script assumes that the $WAS_HOME environment variable has been defined as the IBM WebSphere home directory:

```
## Create the Deployment Manager profile and Deployment Manager, and define
## the Administrator user name and password. serverType must be
## DEPLOYMENT_MANAGER.
${WAS_HOME}/bin/manageprofiles.sh -create -templatePath ${WAS_HOME}
/profileTemplates/management -serverType DEPLOYMENT_MANAGER
-enableAdminSecurity true -adminUserName yourAdminUserName -adminPassword
yourAdminPassword

## Create an empty custom profile. The default custom profile name is Customnn,
## where nn is the next available value in the sequence, starting with 01.
## When you are creating the first custom profile in the cell, the profile name
## is typically Custom01.
${WAS_HOME}/bin/manageprofiles.sh -create -templatePath ${WAS_
HOME}/profileTemplates/managed

## Start the Deployment Manager and federate the custom node on the same machine.
## The startManager.sh command must be issued from the Deployment Manager profile
## directory, Dmgr01. The DepMgrSOAPPort can be determined by opening the file
```

```
## $WAS_HOME/profiles/Dmgr01/logs/AboutThisProfile.txt.
${WAS_HOME}/profiles/Dmgr01/bin/startManager.sh
${WAS_HOME}/bin/addNode.sh localhost DepMgrSAOPPort -user yourAdminUserName
-password yourAdminPassword -profileName CustomProfileName

## Stop the custom node and Deployment Manager.
${WAS_HOME}/profiles/CustomProfileName/bin/stopNode.sh -user yourAdminUserName
-password yourAdminPassword
${WAS_HOME}/profiles/Dmgr01/bin/stopManager.sh -user yourAdminUserName -password
yourAdminPassword
```

## 4.1.2 Environment Setup for IBM WebSphere AS

Before you can use wsadmin to configure your Oracle Fusion Middleware products in
an IBM WebSphere AS environment, you must first create an Application Server
profile. To do so, enter the following commands. This command assumes that the
$WAS_HOME environment variable has been defined as the WebSphere home directory.

- **UNIX:**

  ```
  ## Create the Application Server profile and define the Administrator user
  ## name and password.
  ${WAS_HOME}/bin/manageprofiles.sh -create -templatePath
  ${WAS_HOME}/profileTemplates/default -enableAdminSecurity true
  -adminUserName yourAdminUserName -adminPassword yourAdminPassword
  ```

- **Windows:**

  ```
  WAS_HOME\bin\manageprofiles.cmd -create -templatePath
  WAS_HOME\profileTemplates\default -enableAdminSecurity true
  -adminUserName yourAdminUserName -adminPassword yourAdminPassword
  ```

  In these commands, *WAS_HOME* represents the IBM WebSphere home directory.

## 4.1.3 Prerequisites for DB2 Databases

If you are using a DB2 database, you must perform the steps in this section prior to
using wsadmin to configure FMW products.

> **Note:** As an alternative to these steps, you can use the IBM
> WebSphere Administrator Console to set the DB2_JCC_DRIVER_
> PATH variable to the DB2 driver directory path.

1. On the Deployment Manager machine, run wsadmin, specifying the Deployment
   Manager profile. For example, to run wsadmin for the Dmgr01 profile on a UNIX
   operating system, enter the following command:

   *$MW_HOME*/oracle_common/common/bin/wsadmin.sh -profileName
   Dmgr01

   > **Note:** Before performing Step 2, *do not* execute the
   > OracleMWConfig.startConfig() command to start a
   > configuration session. The following command should be executed
   > outside of a configuration session.

2. Enter the following command at the wsadmin> prompt:

```
AdminTask.setVariable('[-variableName DB2_JCC_DRIVER_PATH
-variableValue Db2DriverDirectory -scope Cell=cellName,
Node=nodeName]')
```

In this command, *Db2DriverDirectory* is the full path to the directory in which the DB2 drivers are located on the Deployment Manager node, *cellName* is the name of the cell, and *nodeName* is the name of the Deployment Manager node.

> **Note:** Ensure that the DB2 JAR files `db2jcc4.jar` and `db2jcc_license_cu.jar` are stored in the *Db2DriverDirectory*.

3. For each Application Server node in the cell, enter the following command at the `wsadmin>` prompt:

```
AdminTask.setVariable('[-variableName DB2_JCC_DRIVER_PATH
-variableValue Db2DriverDirectory -scope Cell=cellName,
Node=nodeName]')
```

In this command, *Db2DriverDirectory* is the full path to the directory in which the DB2 drivers are located on the Application Server node, *cellName* is the name of the cell, and *nodeName* is the name of the Application Server node.

> **Note:** Ensure that the DB2 JAR files `db2jcc4.jar` and `db2jcc_license_cu.jar` are stored in the *Db2DriverDirectory* on the specified Application Server node.

4. Enter the following command to save your changes:

```
AdminConfig.save()
```

5. Enter `exit` to exit from the `wsadmin` session.

Whenever you add a new Application Server node to the cell, repeat this procedure, skipping Step 2.

> **Note:** When configuring data sources for DB2, make sure that you set the `DatabaseType` attribute to `DB2`.

## 4.2  Starting and Ending a Configuration Session

To start a `wsadmin` configuration session, run the `wsadmin` command either in interactive mode or scripting mode as described in "Starting the Oracle Fusion Middleware Wsadmin Command-Line Shell and Connecting to the Deployment Manager" in *Oracle Fusion Middleware Third-Party Application Server Guide*.

Whether executing `wsadmin` commands from the command line or from within a script, you must use the following command to start a new configuration session. This command loads the cell configuration for the Deployment Manager to which you connected.

```
OracleMWConfig.startConfig()
```

After completing the configuration, use the following command to save your configuration changes and end the configuration session:

```
OracleMWConfig.endConfig()
```

> **Caution:** If you do not use the endConfig() command to end your configuration session, any changes you made during the session will not be saved.

To exit the wsadmin utility after completing a configuration session, enter exit.

## 4.3  Adding an Oracle Fusion Middleware Product to a Cell

Example 4–1 demonstrates how to add an Oracle Fusion Middleware product to a cell. To add a product, you select the appropriate product template JAR file. The template JAR files that are used to configure your products in a cell are included with your product installation in the *ORACLE_COMMON_HOME*/common/templates/was directory.

**Example 4–1   Adding a Product to a Cell**

```
# Get the location of the ORACLE_COMMON_HOME directory.
commonHome = java.lang.System.getProperty("COMMON_COMPONENTS_HOME")

# Start Oracle Fusion Middleware product configuration.
OracleMWConfig.startConfig()

# Specify the path to the templates being applied to the cell.
# In this example, the SOA Management template is selected. To specify another
# template, enter the appropriate template name. You can include multiple
# OracleMWConfig.selectTemplate() commands to select multiple templates.
OracleMWConfig.selectTemplate(commonHome + '/common/templates/was/oracle.soa.mgmt_
template_11.1.1.jar')

# Load all the templates that are selected in the current configuration session.
OracleMWConfig.loadTemplates()

# Apply all newly loaded templates and save the changes to the cell.
OracleMWConfig.endConfig()
```

## 4.4  Adding a New Server to a Cell

Example 4–2 demonstrates how to add a new server to a cell. It also demonstrates how to configure and validate the end points on the new server.

**Example 4–2   Adding a New Server to a Cell**

```
# Create a new server.
atts = []
atts.append(['Name','Myserver'])
atts.append(['NodeName','TEST01Node01'])
atts.append(['DevelopmentMode','true'])
newServer = OracleMWConfig.create('Server',atts)

# List the attributes and types supported for the server type.
newServer.show()

[ar-] Name [Myserver]
[ar-] NodeName [TEST01Node01]
[arw] DevelopmentMode [true]
[trw] ListenerAddress
```

```
# Create the listener address
listenerAddress = newServer.create('ListenerAddress',None)

# List the attributes and types supported for the listener address type.
listenerAddress.show()
>>>>> Output:
[trw] AdminListenerAddress
[trw] AdminSSLListenerAddress
[trw] HttpListenerAddress
[trw] HttpsListenerAddress
[trw] SoapListenerAddress
[trw] SibListenerAddress
[trw] SibSSLListenerAddress
[trw] BootstrapListenerAddress

# Create and set ports and hosts for the end points.
atts = []
atts.append(['Host','localhost'])
atts.append(['Port','9000'])
endpoint = listenerAddress.create('HttpListenerAddress',atts)

atts = []
atts.append(['Host','localhost'])
atts.append(['Port','9001'])
endpoint = listenerAddress.create('AdminListenerAddress',atts)

# Validate the ports (check for port conflicts).
OracleMWConfig.validateConfig('InternalPortConflict')
OracleMWConfig.validateConfig('ExternalPortConflict')
```

## 4.5  Configuring Host and Port Settings for End Points

The examples in this section show two methods that you can use to configure the end points on a server.

Example 4–3 demonstrates how to use basic configuration commands to:

- List all available servers
- Display the available listener ports for a server
- Display the current setting for a specific end point
- Change both the host and port for that end point

*Example 4–3   Configuring Host and Port Settings for End Points*

```
# List all available servers.
servers  = OracleMWConfig.getChildren('Server')

# Print all servers.
print servers
>>>>output:
[Server(dmgr), Server(OracleAdminServer), Server(nodeagent), Server(soa_server1),
Server(test), Server(Myserver)]

# Find a server by using name and scope.
adminServer = OracleMWConfig.getChildByName('Server','OracleAdminServer')

# Find the listener addresses.
listenerAddress = adminServer.getChildren('ListenerAddress')
```

```
# Print the listener addresses.
print listenerAddress[0].show()
>>>>output:
[trw] AdminListenerAddress
[trw] AdminSSLListenerAddress
[trw] HttpListenerAddress
[trw] HttpsListenerAddress
[trw] SoapListenerAddress
[trw] SibListenerAddress
[trw] SibSSLListenerAddress
[trw] BootstrapListenerAddress

# Find a specific end point.
adminListenerAddress = listenerAddress[0].getChildren('AdminListenerAddress')

# Print existing end point details.
print adminListenerAddress[0].show()
>>>>output:
[arw] Host [*]
[arw] Port [9001]
[arw] Enable [true]

# Update the host and port for the above end point, and do the same procedure for
# other end points.
adminListenerAddress[0].setValue('Host','localhost')
adminListenerAddress[0].setValue('Port','8000')

# Print the updated end point details.
print adminListenerAddress[0].show()
>>>>output:
[arw] Host [localhost]
[arw] Port [8000]
[arw] Enable [true]
```

Example 4–4 demonstrates how to use basic configuration commands and configuration utility commands to:

- List all available servers

- Find the node name for a specific server

- List the attributes and end points for a specific server and node name

- Get the current host name and port for a specific listener address for a specific server and node name

- Set the host name and port for a specific listener address for a specific server and node name

*Example 4–4   Configuring End Points by Using Utility Commands*

```
# List all available servers.
servers  = OracleMWConfig.getChildren('Server')

# Print all servers.
print servers
>>>>output:
[Server(dmgr), Server(OracleAdminServer), Server(nodeagent), Server(soa_server1)
, Server(test), Server(Myserver)]

# Find a server by using name and scope.
```

```
adminServer = OracleMWConfig.getChildByName('Server','OracleAdminServer')

# Find the node for a server.
print adminServer.getValue('NodeName')
>>>>output:
TEST01Node01

# List the attributes for a server name and node name.
print OracleMWConfigUtilities.showServer('OracleAdminServer','TEST01Node01')
>>>>output:
[ar-] Name [OracleAdminServer]
[ar-] NodeName [myeole-lapNode04]
[arw] DevelopmentMode [false]
[trw] ListenerAddress

# List the end points for a server name and node name.
print
OracleMWConfigUtilities.showEndPointsUsingObject('OracleAdminServer','TEST01Node01
')
>>>>output:
[trw] AdminListenerAddress
[trw] AdminSSLListenerAddress
[trw] HttpListenerAddress
[trw] HttpsListenerAddress
[trw] SoapListenerAddress
[trw] SibListenerAddress
[trw] SibSSLListenerAddress
[trw] BootstrapListenerAddress

# List the host and port of an end point for a specified server name and node
# name.
host =
OracleMWConfigUtilities.getEndPointHostUsingName('OracleAdminServer','TEST01Node01
','AdminListenerAddress')
port =
OracleMWConfigUtilities.getEndPointPortUsingName('OracleAdminServer','TEST01Node01
','AdminListenerAddress')
print host
print port

# Set the host and port of an end point for a specified server name and node
# name.
host =
OracleMWConfigUtilities.setEndPointHostUsingName('OracleAdminServer','TEST01Node01
','HttpListenerAddress','localhost')
port =
OracleMWConfigUtilities.setEndPointPortUsingName('OracleAdminServer','TEST01Node01
','HttpListenerAddress','9001')
```

## 4.6 Creating Servers, Clusters, and Cluster Members

Example 4–5 demonstrates how to:

- Create a new server

- Create a cluster and use the previously created server as the primary cluster member

- Add a new server (cluster member) to the cluster

The new cluster member automatically inherits its attributes from the primary cluster member. For more information about clusters, see Section 1.5, "Creating Clusters and Cloning Application Servers."

**Example 4–5    Creating Servers, Clusters, and Cluster Members**

```
# Create a new server. Note that embedded spaces are not allowed in server names.
#
newServerName1 = 'MyServer1'
node = 'TEST01Node01'
atts = []
atts.append(['Name', newServerName1])
atts.append(['NodeName', node])
atts.append(['DevelopmentMode', 'false'])
newServer = OracleMWConfig.create('Server',atts)
Server: MyServer1 successfully created

# Create a new cluster.
newClusterName = 'MyCluster'
atts = []
atts.append(['Name', newClusterName])
atts.append(['EnableHA', 'true'])
atts.append(['FirstClusterMember', newServerName1])
newCluster = OracleMWConfig.create('ServerCluster', atts)
ServerCluster: MyCluster successfully created

# Create a new server (a new cluster member).
newServerName2 = 'MyServer2'
atts = []
atts.append(['Name', newServerName2])
atts.append(['NodeName', node])
atts.append(['ClusterName', newClusterName])
atts.append(['DevelopmentMode', 'false'])
newServer1 = OracleMWConfig.create('ClusterMember', atts)
ClusterMember: MyServer2 successfully created

# Clone a server
serverName = 'MyServer1'
server_to_clone = OracleMWConfig.getChildByName('Server',serverName)
atts = []
atts.append(['Name','MyServer1c'])
atts.append(['NodeName','TEST01Node01'])
OracleMWConfig.clone(server_to_clone,atts)
Server : MyServer1 successfully cloned as MyServer1c
Server(MyServer1c)
```

## 4.7  Setting File Store Shared Directories for a Cluster

Example 4–6 demonstrates how to:

- Display a list of the available service buses

- Set the file store shared directories for a service bus.

As described in the example comments, there are two ways you can do this.

**Example 4–6    Setting File Store Shared Directories**

```
#
buses = OracleMWConfig.getChildren('SIBus')
busNm = buses[0].getValue('Name')
```

```
# Set the file store directories.
# Use this command or individual commands below to set file store directories.
# OracleMWConfigUtilities.setFileStoreDirectory('LogDirectory', 'Mycluster',
# busNm, '${USER_INSTALL_ROOT}/log')

OracleMWConfigUtilities.setLogDirectory(newClusterName, busNm, '${USER_INSTALL_
ROOT}/log')
OracleMWConfigUtilities.setPermanentStoreDirectory(newClusterName, busNm, '${USER_
INSTALL_ROOT}/permStore')
OracleMWConfigUtilities.setTemporaryStoreDirectory(newClusterName, busNm, '${USER_
INSTALL_ROOT}/tempStore')
```

## 4.8  Creating and Deleting a Data Source

Example 4–7 demonstrates how to create a new Oracle JDBC data source. You can use the same script to create a DB2 data source by specifying 'DB2' as the DatabaseType.

***Example 4–7   Creating a New Data Source***

```
atts = []
atts.append(['Name','myNewDatasource'])
atts.append(['Username','websphere'])
atts.append(['Password','welcome1'])
atts.append(['DatabaseType', 'Oracle'])
# NOTE: Use 'DB2' as the DatabaseType for DB2 data sources.
atts.append(['DatabaseName', 'orcl'])
atts.append(['ServerName', 'example.host.com'])
atts.append(['PortNumber', '1521'])
atts.append(['IsXa', 'true'])
atts.append(['JndiName', 'myJNDI'])
atts.append(['Server', 'soa_server1'])

newDS = OracleMWConfigUtilities.createDatasource(atts)
print newDS.show()
>>>>> Output:
[ar-] Name [myNewDatasource]
[ar-] Type [Normal]
[arw] Username [websphere]
[a-w] Password [********]
[arw] DatabaseName [orcl]
[arw] ServerName [example.host.com]
[arw] PortNumber [1521]
[ar-] URL [jdbc:oracle:thin:@example.host.com:1521/orcl]
[arw] DatabaseType [Oracle]
[arw] DriverType [4]
[arw] IsXa [true]
[arw] IsOracleInstanceType [false]
[arw] IsRac [false]
[ar-] DriverClass [oracle.jdbc.xa.client.OracleXADataSource]
```

Example 4–8 demonstrates how to delete a data source.

***Example 4–8   Deleting a Data Source***

```
dbWrapper = OracleMWConfig.getChildByName('JDBC', 'myNewDatasource')
OracleMWConfig.delete(dbWrapper)
```

## 4.9  Changing a Data Source to an Oracle RAC Data Source

The examples in this section show two methods that you can use to change a standard data source to an Oracle RAC data source.

Example 4–9 demonstrates how to convert a standard data source to an Oracle RAC data source using the convertJdbcNormalDatasourceToRAC command, and then how to display the data source configuration.

***Example 4–9   Converting a Data Source to an Oracle RAC Data Source***

```
hostPortList =
['hostExampleRac1.oracle.com:1521','hostExampleRac2.oracle.com:1234','hostExampleR
ac3.oracle.com:4567']
OracleMWConfigUtilities.convertJdbcNormalDatasourceToRAC('TestNormalDatasource',
'myRACService', hostPortList)
print OracleMWConfigUtilities.showJdbc('TestNormalDatasource')

[ar-] Name [TestNormalDatasource]
[ar-] Type [Normal]
[arw] Username [wlsqa]
[arw] Password [*****]
[arw] DatabaseName [lnrac1]
[arw] ServerName
[hostExampleRac1.oracle.com:hostExampleRac2.oracle.com:hostExampleRac3.oracle.com]
[arw] PortNumber [1521:1234:4567]
[ar-] URL [jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(
HOST=hostExampleRac1.oracle.com)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=hostExa
mpleRac2.oracle.com)(PORT=1234))(ADDRESS=(PROTOCOL=TCP)(HOST=hostExampleRac3.ora
cle.com)(PORT=4567))(FAILOVER=ON)(LOAD_BALANCE=ON))(CONNECT_DATA=(SERVER=DEDICAT
ED)(SERVICE_NAME=myRACService)))]
[arw] DatabaseType [Oracle]
[arw] DriverType [4]
[arw] IsXa [true]
[arw] IsOracleInstanceType [false]
[arw] IsRac [true]
[ar-] DriverClass [oracle.jdbc.xa.client.OracleXADataSource]
```

Example 4–10 demonstrates how to change a standard data source to an Oracle RAC data source using setValue commands, and then how to display the data source configuration.

***Example 4–10   Changing a Data Source to an Oracle RAC Data Source Using setValue()***

```
testDS = OracleMWConfig.getChildByName('JDBC', 'TestNormalDatasource')
testDS.setValue('DatabaseName', 'lnrac1')
testDS.setValue('Username', 'websphere')
testDS.setValue('Password', 'welcome1')
testDS.setValue('IsXa', 'false')
testDS.setValue('IsRac', 'true')
testDS.setValue('RacService', 'exampleRacService')
testDS.setValue('ServerName', 'hostRac1.oracle.com:hostRac2:hostRac3.oracle.com')
testDS.setValue('PortNumber', '1521:1521:1521')
url_testDS = testDS.getValue('URL')
driverClass = testDS.getValue('DriverClass')
# Show testDS object's configuration settings:
print testDS.show()
>>>>> Output:
[ar-] Name [TestNormalDatasource]
[ar-] Type [Normal]
```

```
[arw] Username [websphere]
[a-w] Password [********]
[arw] DatabaseName [lnrac1]
[arw] ServerName [hostRac1.oracle.com:hostRac2:hostRac3.oracle.com]
[arw] PortNumber [1521:1521:1521]
[ar-] URL [jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(
HOST=hostRac1.oracle.com)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=hostRac2)(PORT
=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=hostRac3.oracle.com)(PORT=1521))(FAILOVER=ON
)(LOAD_BALANCE=ON))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=exampleRacServi
ce)))]

[arw] DatabaseType [Oracle]
[arw] DriverType [4]
[arw] IsXa [false]
[arw] IsOracleInstanceType [false]
[arw] IsRac [true]
[ar-] DriverClass [oracle.jdbc.pool.OracleConnectionPoolDataSource]
```

## 4.10 Changing an Oracle RAC Data Source to a Data Source

Example 4–11 demonstrates how to convert an Oracle RAC data source to a standard
data source and then how to display the resulting configuration of the data source.

*Example 4–11   Changing an Oracle RAC Data Source to a Data Source*

```
OracleMWConfigUtilities.convertJdbcRACToNormalDatasource('TestNormalDatasource',
'MS SQL Server', 'false', 'myhost.com', '1234')
print OracleMWConfigUtilities.showJdbc('TestNormalDatasource')
>>>>> Output:
[ar-] Name [TestNormalDatasource]
[ar-] Type [Normal]
[arw] Username [wlsqa]
[arw] Password [*****]
[arw] DatabaseName [lnrac1]
[arw] ServerName [myhost.com]
[arw] PortNumber [1234]
[ar-] URL [jdbc:sqlserver://myhost.com:1234]
[arw] DatabaseType [MS SQL Server]
[arw] DriverType [4]
[arw] IsXa [false]
[arw] IsOracleInstanceType [false]
[arw] IsRac [false]
[ar-] DriverClass [com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource]
```

## 4.11 Configuring a Component Schema

This section demonstrates two methods that you can use to configure a component
schema, validate it, and display its configuration.

Example 4–12 demonstrates how to use utility commands to configure the values for a
component schema. The `OracleMWConfigUtilities.validateJdbc()`
command verifies that all required parameters for component schema configuration
were specified.

*Example 4–12   Configuring a Component Schema By Using Utility Commands*

```
OracleMWConfigUtilities.setJdbcDriverVendor('SOA Infrastructure', 'Oracle')
OracleMWConfigUtilities.setJdbcIsOracleInstanceType('SOA Infrastructure', 'true')
OracleMWConfigUtilities.setJdbcUsername('SOA Infrastructure', 'websphere')
```

```
OracleMWConfigUtilities.setJdbcPassword('SOA Infrastructure', 'welcome1')
OracleMWConfigUtilities.setJdbcDatabaseName('SOA Infrastructure', 'orcl')
OracleMWConfigUtilities.setJdbcServerName('SOA Infrastructure',
'dbhost.example.com')
OracleMWConfigUtilities.setJdbcPortNumber('SOA Infrastructure', '12345')
OracleMWConfigUtilities.validateJdbc('SOA Infrastructure')
url_soaSchemaComp = OracleMWConfigUtilities.getJdbcURL('SOA Infrastructure')
print OracleMWConfigUtilities.showJdbc('SOA Infrastructure')
>>>>> Output:
[ar-] Name [SOA Infrastructure]
[ar-] Type [Schema]
[arw] SchemaOwner [websphere]
[arw] SchemaPassword [********]
[arw] DatabaseName [orcl]
[arw] ServerName [dbhost.example.com]
[arw] PortNumber [12345]
[ar-] URL [jdbc:oracle:thin:@dbhost.example.com:12345:orcl]
[arw] DatabaseType [Oracle]
[arw] DriverType [4]
[arw] IsOracleInstanceType [true]
[arw] IsRac [false]
[ar-] DriverClass [oracle.jdbc.pool.OracleConnectionPoolDataSource]
```

Example 4–13 demonstrates how to use setValue commands to configure a component schema. The soaSchemaComp.validate() call is a standard Oracle configuration API.

***Example 4–13   Configuring a Component Schema Using setValue()***

```
soaSchemaComp = OracleMWConfig.getChildByName('JDBC', 'SOA Infrastructure')
soaSchemaComp.setValue('DatabaseType', 'Oracle')
# Use 'DB2' as the DatabaseType for DB2 data sources
soaSchemaComp.setValue('IsOracleInstanceType', 'true')
soaSchemaComp.setValue('SchemaOwner', 'websphere')
soaSchemaComp.setValue('SchemaPassword', 'welcome1')
soaSchemaComp.setValue('DatabaseName', 'orcl')
soaSchemaComp.setValue('ServerName', 'dbhost.example.com')
soaSchemaComp.setValue('PortNumber', '1521')
soaSchemaComp.validate()
url_soaSchemaComp=soaSchemaComp.getValue('URL')
# Show soaSchemaComp's configuration settings.
print soaSchemaComp.show()
>>>>> Output:
[ar-] Name [SOA Infrastructure]
[ar-] Type [Schema]
[arw] SchemaOwner [websphere]
[a-w] SchemaPassword [********]
[arw] DatabaseName [orcl]
[arw] ServerName [dbhost.example.com]
[arw] PortNumber [1521]
[ar-] URL [jdbc:oracle:thin:@dbhost.example.com:1521:orcl]

[arw] DatabaseType [Oracle]
[arw] DriverType [4]
[arw] IsOracleInstanceType [true]
[arw] IsRac [false]
[ar-] DriverClass [oracle.jdbc.pool.OracleConnectionPoolDataSource]
```

## 4.12 Federating Remote Nodes to a Cell

This section describes the steps that you must perform in either a UNIX or Windows environment to add a remote node to a cell. After adding the remote node, you can use `wsadmin` commands interactively or a `wsadmin` script to perform optional configuration tasks on the remote node.

Prior to adding the node, you must set up the environment on the Deployment Manager machine as described in Section 4.1.1, "Environment Setup for IBM WebSphere ND."

**UNIX**

To add a remote node to an existing cell:

1.  On the Deployment Manager machine, enter the following `wsadmin` command to use a Jython script containing the appropriate `OracleMWConfig` and `OracleMWConfigUtilities` commands to apply product templates to the cell and perform other optional configuration (such as adding servers and clusters):

    `ORACLE_HOME/common/bin/wsadmin.sh -f script_name`

    Replace `ORACLE_HOME` with the directory in which you installed your Oracle products and replace `script_name` with the full path and file name of your Jython script.

2.  On the Deployment Manager machine, enter the following command to start the Deployment Manager:

    `${WAS_HOME}/profiles/profile_name/bin/startManager.sh`

    Replace `profile_name` with the name of the Deployment Manager profile (for example, `Dmgr01`).

3.  View the following file to determine the Deployment Manager host name and SOAP port number. You need this information to connect to the Deployment Manager from the remote node:

    `WAS_HOME/profiles/profile_name/AboutThisProfile.txt`

    Replace `WAS_HOME` with the IBM WebSphere home directory and replace `profile_name` with the name of the Deployment Manager profile.

After completing these steps, perform the following steps on the remote machine:

1.  Enter the following command to create an empty custom profile:

    `${WAS_HOME}/bin/manageprofiles.sh -create -templatePath ${WAS_HOME}/profileTemplates/managed`

    > **Note:** Before executing the `addNode.sh` command in the next step, if you have already applied the Oracle JRF for WebSphere template to the cell, you must shut down and restart the Deployment Manager from the Deployment Manager machine, using `stopManager.sh` and `startManager.sh`.

2.  Enter the following `wsadmin` command to federate (add) the remote node to the cell:

    `${WAS_HOME}/bin/addNode.sh depMgrHost SOAPport -user adminUserName -password adminPassword`

Replace *depMgrHost* with the host name of the Deployment Manager machine, replace *SOAPport* with the SOAP port in the Deployment Manager `AboutThisProfile.txt` file, and replace *adminUserName* and *adminPassword* with the username and password you configured for the Deployment Manager.

3. Enter the following `wsadmin` command to use a Jython script containing the appropriate `OracleMWConfig` and `OracleMWConfigUtilities` commands to apply additional product templates (if needed) to the cell and perform other optional configuration (such as adding servers, clusters, and cluster members):

   `ORACLE_HOME/common/bin/wsadmin.sh -connType SOAP -host hostName -port SOAPport -user adminUserName -password adminPassword -f script_name`

   Replace *hostName* with the Deployment Manager host name and replace *SOAPport* with the SOAP port in the Deployment Manager `AboutThisProfile.txt` file.

**Windows**

To add a remote node in to an existing cell:

1. On the Deployment Manager machine, enter the following `wsadmin` command to use a Jython script containing the appropriate `OracleMWConfig` and `OracleMWConfigUtilities` commands to apply product templates to the cell and perform other optional configuration (such as adding servers and clusters):

   `ORACLE_HOME\oracle_common_home\common\bin\wsadmin.cmd -f script_name`

   Replace *ORACLE_HOME* with the directory in which you installed your Oracle products and replace *script_name* with the full path of your Jython script.

2. On the Deployment Manager machine, enter the following command to start the Deployment Manager:

   `WAS_HOME\profiles\profile_name\bin\startManager.cmd`

   In this directory path, *WAS_HOME* is the IBM WebSphere home directory and *profile_name* the name of the Deployment Manager profile (for example, `Dmgr01`).

3. View the following file to determine the Deployment Manager host name and SOAP port number. You need this information to connect to the Deployment Manager from the remote node:

   `WAS_HOME\profiles\profile_name\AboutThisProfile.txt`

   Replace *profile_name* with the name of the Deployment Manager profile.

After completing these steps, perform the following steps on the remote machine:

1. Enter the following command to create an empty custom profile:

   `WAS_HOME\bin\manageprofiles.cmd -create -templatePath WAS_HOME\profileTemplates\managed`

   > **Note:** Before executing the `addNode.cmd` command in the next step, if you have already applied the Oracle JRF for WebSphere template to the cell, you must shut down and restart the Deployment Manager, using `stopManager.cmd` and `startManager.cmd`.

2. Enter the following `wsadmin` command to federate (add) the remote node to the cell:

*WAS_HOME*\bin\addNode.cmd *depMgrHost SOAPport* -user *adminUserName* -password *adminPassword*

Replace *depMgrHost* with the host name of the Deployment Manager machine, replace *SOAPport* with the SOAP port in the Deployment Manager `AboutThisProfile.txt` file, and replace *adminUserName* and *adminPassword* with the Administrator user name and password that you defined for the Deployment Manager when setting up the environment.

3. Enter the following `wsadmin` command to use a Jython script containing the appropriate `OracleMWConfig` and `OracleMWConfigUtilities` commands to apply additional product templates (if needed) to the cell and perform other optional configuration (such as adding servers, clusters, and cluster members):

*ORACLE_HOME*\common\bin\wsadmin.cmd -connType SOAP -host *hostName* -port *SOAPport* -user *adminUserName* -password *adminPassword* -f *script_name*

Replace *hostName* with the Deployment Manager host name, replace *SOAPport* with the SOAP port in the Deployment Manager `AboutThisProfile.txt` file, replace *adminUserName* and *adminPassword* with the Administrator user name and password that you defined for the Deployment Manager when setting up the environment, and replace *script_name* with the full path of your Jython script.

# 5

# wsadmin Command Reference

This chapter describes all of the currently available Oracle configuration (`OracleMWConfig`) and Oracle configuration utility (`OracleMWConfigUtilities`) commands that you can use with the `wsadmin` command-line utility. These Oracle-specific commands allow you to create, configure, and modify cells for Oracle Fusion Middleware products, as well as add resources to or delete resources from those cells.

This chapter contains the following sections:

- OracleMWConfig Commands
- OracleMWConfigUtilities Commands

In addition to the `wsadmin` commands described in this chapter, you can also use `wsadmin` commands for Oracle Fusion Middleware products that are supported on IBM WebSphere. The commands are identical to the WebLogic Scripting Tool (WLST) commands for these Oracle Fusion Middleware products, although not all of these commands are supported in an IBM WebSphere environment. For more information on these commands, see the following sections in the *Oracle Fusion Middleware Third-Party Application Server Guide*:

- "Using the Oracle Fusion Middleware wsadmin Commands"
- "Using the Web Services wsadmin Commands"
- "Configuring Oracle Fusion Middleware Logging on IBM WebSphere"
- "Configuring MetaData Services (MDS) on IBM WebSphere"
- "Setting Up the Diagnostic Framework"
- "Oracle SOA Suite wsadmin and WLST Command Differences"

See also the appropriate product chapter in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* or refer to the `wsadmin` help.

## 5.1 OracleMWConfig Commands

This section describes the `OracleMWConfig` commands that you can use with the `wsadmin` utility to create, configure, and modify cells.

Table 5–1 lists these `OracleMWConfig` commands.

***Table 5–1  OracleMWConfig Commands***

| Command | Description |
|---------|-------------|
| assign | Assigns the specified application, library, service, or server to the specified target |
| clone | Creates a new server by cloning the specified server object |
| copyFile | Copies a file or directory to a location within the IBM WebSphere configuration repository |
| create | Creates a wrapper object of the specified type with the specified list of attributes |
| delete | Deletes the specified wrapper object |
| dumpStack | (Interactive mode only) Displays a stack trace, if one exists, for the last executed command |
| endConfig | Ends the configuration session |
| getChildByName | Returns a wrapper object of the specified type, name, and scope |
| getChildren | Returns a list of wrapper objects of the specified type |
| getTypes | Returns a list of the supported element types |
| getValue | Returns the value for the specified attribute |
| help | Displays a list of all `OracleMWConfig` commands or displays help for a specified `OracleMWConfig` command |
| list | Lists the names of all elements of the specified type, which must be an assignable type |
| loadTemplates | Loads all the currently selected templates for processing |
| registerOracleHomes | Registers Oracle home directories on local and remote nodes |
| retrieveObject | Retrieves the value corresponding to the specified key from the configuration session's `ObjectStore` |
| selectTemplate | Selects an Oracle Fusion Middleware configuration template |
| setValue | Sets the value for the specified attribute |
| showTemplates | Shows applied and selected templates |
| startConfig | Starts the configuration session |
| storeObject | Stores the specified value with the specified key in the configuration session's `ObjectStore` for later use in the session |
| suppressException | (Interactive mode only) Suppresses exceptions for a cleaner display |
| unassign | Unassigns the specified application, service, or server from the specified target |
| validateConfig | Validates the configuration according to the specified option |

## 5.1.1  assign

Assigns the specified application, library, custom service, or server to a server or cluster object.

### Syntax

```
OracleMWConfig.assign(type, name, wrapper, atts = [])
```

| Argument | Definition |
|----------|------------|
| type | The type of element to be assigned. Valid types are Server, Library, CustomService, and Application. |
| name | The name of the element to be assigned |
| wrapper | The wrapper object of the target (a server or cluster, depending on the type being assigned) |
| atts | A list of attributes of the element to be assigned. Use attributes to clarify the operation; for example, with server to cluster assignment, you can specify the node. |

**Example**

The following example assigns the server server1 to the cluster cluster1.

```
serverName = 'server1'
clusterName = 'cluster1'
cluster1 = OracleMWConfig.getChildByName('ServerCluster', clusterName)
OracleMWConfig.assign('Server', serverName, cluster1)
```

### 5.1.2 clone

Creates a new server by cloning the specified server object. The specified attributes define the server name and node name for the cloned server.

**Syntax**

```
OracleMWConfig.clone(serverObject, atts)
```

| Argument | Definition |
|----------|------------|
| serverObject | The server object to clone |
| atts | A list of attributes. Valid attributes are Name and NodeName. |

**Example**

The following example clones a new server (server2) from an existing server (server1) and sets the node name for the new server to TESTNode03.

```
serverName = 'server1'
server = OracleMWConfig.getChildByName('Server', serverName)
atts = []
atts.append(['Name','server2'])
atts.append(['NodeName','TESTNode03'])
OracleMWConfig.clone(server, atts)
```

### 5.1.3 copyFile

Copies a file or directory to a location within the IBM WebSphere configuration repository. For information on this repository, refer to your IBM WebSphere documentation, specifically the documentation for AdminConfig.createDocument and related commands.

**Syntax**

```
OracleMWConfig.copyFile(src, scope, wasRelativePath,
configGroup)
```

| Argument | Definition |
|---|---|
| src | The path of the file or directory to be copied |
| scope | One of these values: cell, node, cluster, or server |
| wasRelativePath | The destination path, which is relative to one of the following paths, depending on the specified scope:<br><br>■ If scope is cell:<br><br>*profileName*/config/cells/*cellName*/oracle<br><br>■ If scope is node:<br><br>*profileName*/config/cells/*cellName*/nodes/*nodeName*<br>*e*<br><br>■ If scope is cluster:<br><br>*profileName*/config/cells/*cellName*/clusters/*clusterName*<br><br>■ If scope is server:<br><br>*profileName*/config/cells/*cellName*/nodes/*nodeName*<br>*e*/servers/*serverName*<br><br>In these paths, *profileName* is the name of the Deployment Manager profile for the current wsadmin session, *cellName* is the name of the cell, *nodeName* is the name of the current node, *clusterName* is the name of the current cluster, and *serverName* is the name of the current server. |
| configGroup | Either None or a server group name that is defined in config-groups.xml. This is ignored if the scope is cell. Server group names are defined in the <domain-topology> element of config-groups.xml, which is located in the following directory:<br><br>*WAS_*<br>*HOME*/profiles/*profileName*/config/cells/*cellName*/or<br>acle/init-info/ |

### Example

This example copies the compDomainDir directory to the fmwconfig directory.

```
OracleMWConfig.copyFile(compDomainDir, 'cell', 'fmwconfig', None)
```

This example copies the CompServerDir directory to the fmwconfig directory.

```
OracleMWConfig.copyFile(compServerDir, 'server', 'fmwconfig', 'JRF-ADMIN-SVR')
```

## 5.1.4 create

Creates a wrapper object of the specified type with the specified list of attributes.

### Syntax

```
OracleMWConfig.create(type, atts)
```

| Argument | Definition |
|---|---|
| type | The type of element to be created. Use the getTypes command to list valid types. |
| atts | A list of attributes |

**Example**

The following example creates a server wrapper object (newServer) and a JDBC data source wrapper object (newDS).

```
atts = []
atts.append(['Name','Myserver'])
atts.append(['NodeName','TEST01Node03'])
atts.append(['DevelopmentMode','true'])
newServer = OracleMWConfig.create('Server',atts)

Server : Myserver successfully created

jdbcAtts = []
jdbcAtts.append(['Name','myNewDatasource'])
jdbcAtts.append(['Username','websphere'])
jdbcAtts.append(['Password','welcome1'])
jdbcAtts.append(['DatabaseType', 'Oracle'])
jdbcAtts.append(['DatabaseName', 'orcl'])
jdbcAtts.append(['ServerName', 'example.host.com'])
jdbcAtts.append(['PortNumber', '1521'])
jdbcAtts.append(['IsXa', 'true'])
jdbcAtts.append(['StatementCacheSize', '1024'])
jdbcAtts.append(['AgedTimeout', '12345'])
jdbcAtts.append(['datasourceCustomKey', 'datasourceCustomValue'])
jdbcAtts.append(['JndiName', 'myJNDI'])
jdbcAtts.append(['Server', 'soa_server1'])
newDS = OracleMWConfig.create('JDBC', jdbcAtts)

JDBC: myNewDatasource successfully created
```

## 5.1.5 delete

Deletes the specified wrapper object.

**Syntax**

```
OracleMWConfig.delete(wrapper)
```

| Argument | Definition |
|----------|------------|
| wrapper | The wrapper object to delete |

**Examples**

The following example deletes the Administration Server (OracleAdminServer).

```
server = OracleMWConfig.getChildByName('Server','OracleAdminServer')
OracleMWConfig.delete(server)
```

The following example deletes the SOA Infrastructure data source.

```
dbinfo = OracleMWConfig.getChildByName('JDBC','SOA Infrastructure')
OracleMWConfig.delete(dbinfo)
```

## 5.1.6 dumpStack

This command can be used only in interactive mode; it cannot be used in scripts. It displays a stack trace, if one exists, for the last wsadmin command that was executed. See suppressException for information about how to prevent exceptions from displaying in the console when an error occurs.

**Syntax**

```
OracleMWConfig.dumpStack()
```

**Example**

The following example shows some of the dumpStack output after specifying an incorrect value for a type argument:

```
OracleMWConfig.dumpStack()

com.oracle.cie.was.wsadmin.WsadminException:
com.oracle.cie.was.wsadmin.UnsupportedTypeException: CFGFWK-68049:  The requested
type Sever is not supported.
        at
com.oracle.cie.was.wsadmin.WsadminState.processCommand(WsadminState.java:193)
.
.
.
        ... 59 more
```

## 5.1.7 endConfig

Ends the configuration session by running the relevant wsadmin commands, including a call to AdminConfig.save() for saving all the changes. After this command is executed, it may take several minutes for the configuration changes to be written to the cell.

**Syntax**

```
OracleMWConfig.endConfig()
```

**Example**

The following example shows a successful save of the cell configuration.

```
OracleMWConfig.endConfig()

...
CFGFWK-24018:  Successfully saved changes to WebSphere configuration.
```

## 5.1.8 getChildByName

Returns a wrapper object of the specified type and name if the scope is not specified; otherwise, returns a wrapper object of the specified type and name within the specified scope.

**Syntax**

```
OracleMWConfig.getChildByName(type,name,scope)
```

| Argument | Definition |
| --- | --- |
| type | The child element type. Use the getTypes command to list valid types. |
| name | The child element name |
| scope | Optional. The scope, which can be a server name, cluster name, node name, or cell name. For example, you may need to specify the node name if a specified server name exists on different nodes. |

**Examples**

The following example returns a wrapper object for the OracleAdminServer.

```
server = OracleMWConfig.getChildByName('Server','OracleAdminServer')
```

The following example returns a wrapper object for the OracleAdminServer on Test01Node03.

```
server =
OracleMWConfig.getChildByName('Server','OracleAdminServer','TEST01Node03')
```

### 5.1.9 getChildren

Returns a list of wrapper objects of the specified type.

**Syntax**

```
OracleMWConfig.getChildren(type)
```

| Argument | Definition |
|----------|------------|
| type | The child element type. Use the getTypes command to list valid types |

**Example**

This example lists all servers:

```
servers  = OracleMWConfig.getChildren('Server')
print servers

Server(dmgr), Server(OracleAdminServer), Server(nodeagent), Server(soa_server1),
Server(test), Server(Myserver)
```

### 5.1.10 getTypes

Returns a subset of the child elements of `was-config` from the `config-was` schema, plus JDBC.

**Syntax**

```
OracleMWConfig.getTypes()
```

**Example**

This example shows the output of `getTypes()`.

```
OracleMWConfig.getTypes()
array(['SIBus', 'Server', 'ServerCluster', 'JDBC', 'ClusterMember'],
java.lang.String)
```

### 5.1.11 getValue

Returns the value for the specified attribute. See setValue for related information.

**Syntax**

```
Object.getValue(attributeName)
```

| Argument | Definition |
|----------|------------|
| attributeName | The name of the attribute whose value you want to retrieve |

**Example**

This example retrieves the node name for the adminServer object, which is defined as the Administration Server (OracleAdminServer).

```
adminServer = OracleMWConfig.getChildByName('Server','OracleAdminServer')
print adminServer.getValue('NodeName')

Node01
```

## 5.1.12  help

Displays help information for the specified OracleMWConfig command. If you omit the argument, this command displays the list of OracleMWConfig commands.

For more information about displaying help, see "Using the Oracle Fusion Middleware wsadmin Command-Line Online Help" in *Oracle Fusion Middleware Third-Party Application Server Guide*.

**Syntax**

```
OracleMWConfig.help(topic)
```

| Argument | Definition |
| --- | --- |
| topic | The command for which you want to display help |

**Example**

The following command displays the help for the OracleMWConfig.copyFile command:

```
print OracleMWConfig.help('copyFile')

Copies a file or directory to a location within WebSphere's configuration
repository.

Syntax:

OracleMWConfig.copyFile(src, scope, wasRelativePath, configGroup)
src - path of the file or directory to be copied.
scope - one of these values: 'cell', 'node', 'cluster' or 'server'.
wasRelativePath - destination path, which is relative to the 'oracle'
directory within the WebSphere profile.
configGroup - a server group name defined in config-groups.xml.  This is
ignored if the scope is 'cell'.

Example:

OracleMWConfig.copyFile(compDomainDir, 'cell', 'fmwconfig',
None)OracleMWConfig.copyFile(compServerDir, 'server', 'fmwconfig',
'JRF-ADMIN-SVR')
```

## 5.1.13  list

Lists the names of all elements of the specified type, which must be an assignable type.

**Syntax**

```
OracleMWConfig.list(assignableType)
```

| Argument | Definition |
|---|---|
| assignableType | An assignable type whose elements are to be listed. Valid types are: |
| | ■ Application |
| | ■ Library |
| | ■ CustomService |
| | ■ Datasource |
| | ■ ResourceAdapter |
| | ■ Queue |
| | ■ ConnectionFactory |
| | ■ TopicConnectionFactory |
| | ■ URLProvider |
| | ■ WorkManager |
| | ■ BusMember |
| | ■ Topic |
| | ■ ActivationSpec |

**Example**

The following example lists all application elements in the cell.

```
OracleMWConfig.list('Application')

array(['wsil-wls', 'FMW Welcome Page Application_11.1.0.0.0'], java.lang.String)
```

### 5.1.14 loadTemplates

Loads all of the currently selected templates for processing. Run this command after using selectTemplate to select the templates to load.

**Syntax**

```
OracleMWConfig.loadTemplates()
```

**Example**

The following example shows the output for the loadTemplate command, where one template (Oracle JRF:1.1.0.0) had been selected for loading.

```
OracleMWConfig.loadTemplates()

CFGFWK-24011:  Added template Oracle JRF:1.1.0.0.
```

### 5.1.15 registerOracleHomes

Registers Oracle home directories on local and remote nodes. This command should be run at least once from each host system corresponding to an IBM WebSphere node within the cell.

**Syntax**

```
OracleMWConfig.registerOracleHomes()

OracleMWConfig.registerOracleHomes(dmgrProfilePath)
```

| Argument | Definition |
|---|---|
| dmgrProfilePath | The path to a Deployment Manager profile on the current node |

**Examples**

The following example registers all Oracle home directories as part of the configuration session.

```
OracleMWConfig.registerOracleHomes()
```

The following example registers all Oracle home directories in offline mode, independent of the configuration session.

```
OracleMWConfig.registerOracleHomes(dmgrProfilePath)
```

## 5.1.16 retrieveObject

Retrieves the value corresponding to the specified key from the configuration session's `ObjectStore`. The key-value pair must have been previously stored using the storeObject command.

**Syntax**

```
OracleMWConfig.retrieveObject(key)
```

| Argument | Definition |
|---|---|
| key | The name of the key whose value you want to retrieve |

**Example**

The following example defines the value for the Admin key, then retrieves the value based on the key name.

```
adminServer = OracleMWConfig.getChildByName('Server','OracleAdminServer')
OracleMWConfig.storeObject('Admin','adminServer')
OracleMWConfig.retrieveObject('Admin')

adminServer
```

## 5.1.17 selectTemplate

Selects an Oracle Fusion Middleware configuration template. For information about Oracle Fusion Middleware templates, see Section 1.2.1, "Oracle Fusion Middleware Product Templates."

**Syntax**

```
OracleMWConfig.selectTemplate(path)
```

| Argument | Definition |
|---|---|
| path | The path of the template JAR file |

**Example**

```
commonHome = java.lang.System.getProperty("COMMON_COMPONENTS_HOME")
OracleMWConfig.selectTemplate(commonHome + '/common/templates/was/jrf_template_
11.1.1.jar')
```

```
CFGFWK-24010:  Selecting template and all its prerequisites.
CFGFWK-24014:  Currently selected templates are
               Oracle JRF:1.1.0.0
```

## 5.1.18 setValue

Sets the value for the specified attribute.

### Syntax

*Object*.setValue(attributeName, attributeValue)

| Argument | Definition |
|----------|------------|
| attributeName | The name of the attribute whose value you want to set |
| attributeValue | The value to which you want to set the attribute |

### Example

The following example displays the attributes for listenerAddress, displays the current settings for adminListenerAddress, and then sets a new Host value for adminListenerAddress.

```
adminServer = OracleMWConfig.getChildByName('Server', 'OracleAdminServer')
listenerAddress = adminServer.getChildren('ListenerAddress')
print listenerAddress[0].show()

[trw] AdminListenerAddress
[trw] AdminSSLListenerAddress
[trw] HttpListenerAddress
[trw] HttpsListenerAddress
[trw] SoapListenerAddress
[trw] SibListenerAddress
[trw] SibSSLListenerAddress
[trw] BootstrapListenerAddress
adminListenerAddress = listenerAddress[0].getChildren('AdminListenerAddress')
print adminListenerAddress[0].show()

[arw] Host [*]
[arw] Port [9001]
[arw] Enable [true]
adminListenerAddress[0].setValue('Host','newhost.domain.com')
```

## 5.1.19 showTemplates

Shows Oracle Fusion Middleware configuration templates that were previously applied or are currently selected.

### Syntax

OracleMWConfig.showTemplates()

### Example

This example shows that there are three previously applied templates and no currently selected templates.

```
print OracleMWConfig.showTemplates()

CFGFWK-24023:  Previously applied templates:
```

```
  Oracle CIE Core:11.1.1.0
  Oracle JRF:11.1.1.0
  Oracle SOA Suite:11.1.1.0
CFGFWK-24024:  Currently selected templates:
  None
```

## 5.1.20  startConfig

Starts the configuration session by reading relevant data from the cell. You may have to wait several minutes for the cell configuration to load.

**Syntax**

```
OracleMWConfig.startConfig()
```

**Example**

This example shows the output for `startConfig` when the cell is successfully loaded.

```
OracleMWConfig.startConfig()

CFGFWK-24012:  Loaded cell configuration.
```

## 5.1.21  storeObject

Stores the specified value with the specified key in the configuration session's `ObjectStore` for later use in the session. The value for a key can be retrieved using `retrieveObject`.

**Syntax**

```
OracleMWConfig.storeObject(key, value)
```

| Argument | Definition |
| --- | --- |
| key | The object key |
| value | The value to assign to this key |

**Example**

The following example creates the `adminServer` object, and then stores this object as `Admin`.

```
adminServer = OracleMWConfig.getChildByName('Server','OracleAdminServer')
OracleMWConfig.storeObject('Admin','adminServer')
```

## 5.1.22  suppressException

Use this command only in interactive mode. It suppresses exceptions for a cleaner display. See Section 5.1.6, "dumpStack," for information about how to retrieve error information when suppressing exceptions. When exceptions are suppressed, the exceptions are cached and can be retrieved using `dumpStack()`. The exception cache is reset for each `wsadmin` command that is executed.

**Syntax**

```
OracleMWConfig.suppressExceptions(suppressExceptions)
```

| Argument | Definition |
|----------|------------|
| `suppressExceptions` | Set this value to 1 to suppress exceptions. Set this value to 0 to display exceptions. |

### Example

The following example prevents exceptions from being displayed in the console when an error occurs.

```
suppressExceptions = 1
OracleMWConfig.suppressExceptions(suppressExceptions)
```

## 5.1.23 unassign

Unassigns the specified application, service, library, or server from the specified target. See assign for related information.

### Syntax

```
OracleMWConfig.unassign(type, name, wrapper, atts = [])
```

| Argument | Definition |
|----------|------------|
| `type` | The type of element to be unassigned. Valid types are:<br><br>■  `Server`<br><br>■  `Library`<br><br>■  `CustomService`<br><br>■  `Application` |
| `name` | The name of the element to be unassigned |
| `wrapper` | The wrapper object (a server or cluster, depending on the type being unassigned) |
| `atts` | An optional list of attributes of the element to be unassigned |

### Example

The following example unassigns the server `server1` from the cluster `cluster1`.

```
serverName = 'server1'
clusterName = 'cluster1'
cluster1 = OracleMWConfig.getChildByName('ServerCluster', clusterName)
OracleMWConfig.unassign('Server', serverName, cluster1)
```

## 5.1.24 validateConfig

Validates the configuration according to the specified option.

### Syntax

```
OracleMWConfig.validateConfig(optionName)
```

| Argument | Definition |
|---|---|
| `optionName` | The name of the option. The supported options are `InternalPortConflict` and `ExternalPortConflict`. |
| | `InternalPortConflict` checks for port conflicts on the same host, for each host in the cell. For example, if you have two host machines in the cell, it checks for conflicts on host1 and then checks for conflicts on host2. |
| | `ExternalPortConflict` checks against active processes on each host. |

**Examples**

The following example checks for internal port conflicts, with a successful result.

```
OracleMWConfig.validateConfig('InternalPortConflict')

validateConfig  InternalPortConflict
succeed: validateConfig  InternalPortConflict
```

The following example checks for external port conflicts, with a successful result.

```
OracleMWConfig.validateConfig('ExternalPortConflict')

validateConfig  ExternalPortConflict
succeed: validateConfig  ExternalPortConflict
```

## 5.2 OracleMWConfigUtilities Commands

This section describes the `OracleMWConfigUtilities` commands that you can use with the `wsadmin` utility to:

- Retrieve nodes

- Configure, modify, and validate end points

- Configure, modify, and delete JDBC and Oracle RAC data sources and component schemas

- Configure and modify file store shared directories.

Table 5–2 lists all of the available `OracleMWConfigUtilities` commands in alphabetic order. The commands are grouped into the following sections:

- help

- Node Functions

- Server and Endpoint Utility Functions

- JDBC Utility Functions

- JDBC Utility Functions for Oracle RAC

- File Store Utility Functions

*Table 5–2    OracleMWConfigUtilities Commands*

| Command | Description |
|---|---|
| convertJdbcNormalDatasourceToRAC | Converts a normal data source or component schema to an Oracle RAC data source |

*Table 5–2   (Cont.) OracleMWConfigUtilities Commands*

| Command | Description |
|---|---|
| convertJdbcRACToNormalDatasource | Converts an Oracle RAC data source to a normal data source or component schema |
| createDatasource | Creates a new data source with the specified data source attribute list |
| deleteDatasource | Deletes the specified data source |
| getEndPointHostUsingName | Returns the value of the `Host` attribute of the specified end point for the specified server name and node name |
| getEndPointHostUsingObject | Returns the value of the `Host` attribute of the specified end point for the specified server |
| getEndPointPortUsingName | Returns the value of the `Port` attribute of the specified end point for the specified server name and node name |
| getEndPointPortUsingObject | Returns the value of the `Port` attribute of the specified end point for the specified server |
| getFileStoreDirectory | Returns the sharable file store directory of the specified type for a cluster name and bus name |
| getJdbcDatabaseName | Returns the value of the `DatabaseName` attribute of the specified data source or component schema |
| getJdbcDatasourceNames | Returns a list of normal JDBC data source instance names |
| getJdbcDriverClass | Returns the value of the `DriverClass` attribute of the specified data source or component schema |
| getJdbcDriverVendor | Returns the value of the `DatabaseType` attribute of the specified data source or component schema |
| getJdbcDriverXa | Returns the Boolean value of the `IsXa` attribute of the specified data source or component schema |
| getJdbcInstanceType | Returns the string `Normal` if the specified instance is a normal data source or returns `Schema` if the specified instance is a component schema |
| getJdbcIsOracleInstanceType | Returns the Boolean value of the `IsOracleInstanceType` attribute of the specified data source or component schema |
| getJdbcPortNumber | Returns the value of the `PortNumber` attribute of the specified data source or component schema |
| getJdbcRacServiceName | Returns the Oracle RAC data source service name of the specified data source or component schema |
| getJdbcSchemaComponentNames | Returns a list of defined component schema instance names |
| getJdbcServerName | Returns the value of the `ServerName` attribute of the specified data source or component schema |
| getJdbcURL | Returns the value of the `URL` attribute of the specified data source or component schema |
| getJdbcUsername | Returns the value of the `Username` attribute of the specified data source or component schema |
| getLocalNode | Retrieves the host of the current machine, then delegates to the `getNodeByHost()` function to retrieve a node value |

*Table 5–2   (Cont.)  OracleMWConfigUtilities Commands*

| Command | Description |
| --- | --- |
| getLogDirectory | Returns the sharable log directory of the specified cluster name and bus name |
| getNodeByHost | Retrieves the first non-Deployment Manager node that has a matching host name value |
| getNodeByServer | Retrieves the node of the first server encountered with the specified name |
| getPermanentStoreDirectory | Returns the sharable permanent store directory of the specified cluster name and bus name |
| getTemporaryStoreDirectory | Returns the sharable temporary store directory of the specified cluster name and bus name |
| help | Provides help for the `OracleMWConfigUtilities` commands |
| setEndPointHostUsingName | Sets the value of the `Host` attribute of the specified end point for the specified server name and node name |
| setEndPointHostUsingObject | Sets the value of the `Host` attribute of the specified end point for the specified server |
| setEndPointPortUsingName | Sets the value of the `Port` attribute of the specified end point for the specified server name and node name |
| setEndPointPortUsingObject | Sets the value of the `Port` attribute of the specified end point for the specified server |
| setFileStoreDirectory | Sets the sharable file store directory of the specified type for the specified cluster name and bus name |
| setJdbcDatabaseName | Sets the value of the `DatabaseName` attribute of the specified data source or component schema |
| setJdbcDriverVendor | Sets the value of the `DatabaseType` attribute of the specified data source or component schema |
| setJdbcDriverXa | Sets the value (true or false) of the `IsXa` attribute of the specified data source or component schema |
| setJdbcIsOracleInstanceType | Sets the Oracle database URL to be instance type (true) or service type (false) for the specified data source or component schema |
| setJdbcPassword | Sets the value of the `Password` attribute of the specified data source or component schema |
| setJdbcPortNumber | Sets the value of the `PortNumber` attribute of the specified data source or component schema |
| setJdbcRacHostsPorts | Sets the Oracle RAC database hosts and ports of the specified Oracle RAC data source or Oracle RAC component schema |
| setJdbcRacServiceName | Sets the Oracle RAC database service name of the specified Oracle RAC data source or Oracle RAC component schema |
| setJdbcServerName | Sets the value of the `ServerName` attribute of the specified data source or component schema |
| setJdbcUsername | Sets the value of the `Username` attribute of the specified data source or component schema |

*Table 5–2   (Cont.)  OracleMWConfigUtilities Commands*

| Command | Description |
| --- | --- |
| setLogDirectory | Sets the sharable log directory for the specified cluster name and bus name |
| setPermanentStoreDirectory | Sets the sharable permanent store directory for the specified cluster name and bus name |
| setTemporaryStoreDirectory | Sets the sharable temporary store directory for the specified cluster name and bus name |
| showEndPointsUsingName | Displays the list of end points of the specified server name and node name |
| showEndPointsUsingObject | Displays the list of end points of the specified server |
| showJdbc | Displays the list of configuration settings of the specified data source or component schema |
| showServer | Displays the child nodes and attributes of the specified server on the specified node |
| validateJdbc | Validates the JDBC data source and component schema data settings |

## 5.2.1  help

Displays help information for the specified `OracleMWConfigUtilities` command. If you omit the argument, this command displays the list of `OracleMWConfigUtilities` commands with descriptions.

For more information about displaying help, see "Using the Oracle Fusion Middleware wsadmin Command-Line Online Help" in *Oracle Fusion Middleware Third-Party Application Server Guide*.

### Syntax

```
OracleMWConfigUtilities.help(topic)
```

| Argument | Definition |
| --- | --- |
| topic | The command for which you want to display help |

### Examples

The following example prints a list of all `OracleMWConfigUtilities` commands with a description for each.

```
print OracleMWConfigUtilities.help()
```

The following example prints the help for the `getEndPointHostUsingName` command.

```
print OracleMWConfigUtilities.help('getEndPointHostUsingName')
```

## 5.2.2  Node Functions

This section describes the following utility functions for retrieving a node:

- getLocalNode
- getNodeByHost

■   getNodeByServer

### 5.2.2.1 getLocalNode

Retrieves the host name of the current machine, and then calls the `getNodeByHost()` function to retrieve a node value.

**Syntax**

```
OracleMWConfigUtilities.getLocalNode()
```

**Example**

The following example shows the output for this command if the local node is Node01.

```
print OracleMWConfigUtilities.getLocalNode()
```

```
Node01
```

### 5.2.2.2 getNodeByHost

Retrieves the first non-Deployment Manager node that has a matching host name value.

**Syntax**

```
OracleMWConfigUtilities.getNodeByHost(hostName)
```

| Argument | Description |
|----------|-------------|
| hostName | The host name of the host from which you want to retrieve the node |

**Example**

The following example displays the node name of the `example.domain.com` host.

```
print OracleMWConfigUtilities.getNodeByHost(example.domain.com)
```

```
Node01
```

### 5.2.2.3 getNodeByServer

Retrieves the node of the first server encountered with the specified name.

**Syntax**

```
OracleMWConfigUtilities.getNodeByServer(serverName)
```

| Argument | Description |
|----------|-------------|
| serverName | The name of the server from which you want to retrieve the node |

**Example**

The following example retrieves the node name of the server soa_server1.

```
print OracleMWConfigUtilities.getNodeByServer('soa_server1')
```

```
Node01
```

## 5.2.3  Server and Endpoint Utility Functions

This section describes the following utility functions, which are used to manage server end points.

- showEndPointsUsingName
- showEndPointsUsingObject
- showServer
- getEndPointHostUsingName
- getEndPointHostUsingObject
- getEndPointPortUsingName
- getEndPointPortUsingObject
- setEndPointHostUsingName
- setEndPointHostUsingObject
- setEndPointPortUsingName
- setEndPointPortUsingObject

### 5.2.3.1  showEndPointsUsingName

Displays the list of end points for the specified server name and node name.

**Syntax**

```
OracleMWConfigUtilities.showEndPointsUsingName(serverName,
nodeName)
```

| Argument | Definition |
|---|---|
| serverName | The name of the server |
| nodeName | The name of the node |

**Example**

The following example displays the list of end points for the server OracleAdminServer on the TEST01Node03 node.

```
print
OracleMWConfigUtilities.showEndPointsUsingName('OracleAdminServer','TEST01Node03')

[trw] AdminListenerAddress
[trw] AdminSSLListenerAddress
[trw] HttpListenerAddress
[trw] HttpsListenerAddress
[trw] SoapListenerAddress
[trw] SibListenerAddress
[trw] SibSSLListenerAddress
[trw] BootstrapListenerAddress
[trw] OrbListenerAddress
[trw] SasSSLServerAuthListenerAddress
[trw] Cisv2SSLServerAuthListenerAddress
[trw] Cisv2SSLMutualAuthListenerAddress
[trw] DcsUnicastListenerAddress
[trw] SipListenerAddress
[trw] SipSSLListenerAddress
[trw] SibMQListenerAddress
```

```
[trw] SibMQSSLListenerAddress
[trw] IpcListenerAddress
[trw] NodeDiscoveryListenerAddress
[trw] NodeMulticastListenerAddress
[trw] NodeIPV6ListenerAddress
```

### 5.2.3.2  showEndPointsUsingObject

Displays the list of end points for the specified server.

**Syntax**

```
OracleMWConfigUtilities.showEndPointsUsingObject(serverObject)
```

| Argument | Definition |
|----------|------------|
| serverObject | The server object |

**Example**

The following example displays the list of end points for the OracleAdminServer server.

```
server = OracleMWConfig.getChildByName('Server','OracleAdminServer')
print OracleMWConfigUtilities.showEndPointsUsingObject(server)

[trw] AdminListenerAddress
[trw] AdminSSLListenerAddress
[trw] HttpListenerAddress
[trw] HttpsListenerAddress
[trw] SoapListenerAddress
[trw] SibListenerAddress
[trw] SibSSLListenerAddress
[trw] BootstrapListenerAddress
[trw] OrbListenerAddress
[trw] SasSSLServerAuthListenerAddress
[trw] Cisv2SSLServerAuthListenerAddress
[trw] Cisv2SSLMutualAuthListenerAddress
[trw] DcsUnicastListenerAddress
[trw] SipListenerAddress
[trw] SipSSLListenerAddress
[trw] SibMQListenerAddress
[trw] SibMQSSLListenerAddress
[trw] IpcListenerAddress
[trw] NodeDiscoveryListenerAddress
[trw] NodeMulticastListenerAddress
[trw] NodeIPV6ListenerAddress
```

### 5.2.3.3  showServer

Displays the child nodes and attributes for the specified server on the specified node.

**Syntax**

```
OracleMWConfigUtilities.showServer(serverName, nodeName)
```

| Argument | Definition |
|----------|------------|
| serverName | The name of the server |
| nodeName | The name of the node |

**Example**

The following example displays the attributes for the myServer server on the TEST01Nod03 node.

```
OracleMWConfigUtilities.showServer('myServer', 'TEST01Node03')

[ar-] Name [soa_server1]
[ar-] NodeName [adc6140173Node02]
[arw] DevelopmentMode [false]
[trw] ListenerAddress
```

### 5.2.3.4  getEndPointHostUsingName

Returns the value of the `Host` attribute of the specified end point for the specified server name and node name.

**Syntax**

```
OracleMWConfigUtilities.getEndPointHostUsingName(serverName,
nodeName, endPointName)
```

| Argument | Definition |
|----------|------------|
| serverName | The name of the server |
| nodeName | The name of the node |
| endPointName | The name of the end point |

**Example**

The following example returns the value of the `Host` attribute for the `SoapListenerAddress` end point on the OracleAdminServer server on node TEST01Node03.

```
host = OracleMWConfigUtilities.getEndPointHostUsingName
('OracleAdminServer','TEST01Node03','SoapListenerAddress')
print host

myserver.domain.com
```

### 5.2.3.5  getEndPointHostUsingObject

Returns the value of the `Host` attribute of the specified end point for the specified server.

**Syntax**

```
OracleMWConfigUtilities.getEndPointHostUsingObject(serverObject,
endPointName)
```

| Argument | Definition |
|----------|------------|
| serverObject | The server object |
| endPointName | The name of the end point |

**Example**

The following example returns the value of the `Host` attribute for the `SoapListenerAddress` end point for the OracleAdminServer server.

```
server = OracleMWConfig.getChildByName('Server','OracleAdminServer')
```

```
print
OracleMWConfigUtilities.getEndPointHostUsingObject(server,'SoapListenerAddress')
```

```
'myserver.domain.com'
```

### 5.2.3.6 getEndPointPortUsingName

Returns the value of the `Port` attribute of the specified end point for the specified server name and node name.

**Syntax**

```
OracleMWConfigUtilities.getEndPointPortUsingName(serverName,
nodeName,endPointName)
```

| Argument | Definition |
|---|---|
| serverName | The name of the server |
| nodeName | The name of the node |
| endPointName | The name of the end point |

**Example**

The following example returns the value of the `Port` attribute for the `SoapListenerAddress` end point for the OracleAdminServer server name on the TEST01Node01 node.

```
print
OracleMWConfigUtilities.getEndPointPortUsingName('OracleAdminServer','TEST01Node01
','SoapListenerAddress')
```

```
8882
```

### 5.2.3.7 getEndPointPortUsingObject

Returns the value of the `Port` attribute of the specified end point for the specified server.

**Syntax**

```
OracleMWConfigUtilities.getEndPointPortUsingObject(server,
endPointName)
```

| Argument | Definition |
|---|---|
| server | The server object |
| endPointName | The name of the end point |

**Example**

The following example returns the value of the `Port` attribute for the `SoapListenerAddress` end point on the OracleAdminServer server.

```
server = OracleMWConfig.getChildByName('Server','OracleAdminServer')
print
OracleMWConfigUtilities.getEndPointHostUsingObject(server,'SoapListenerAddress')
```

```
8882
```

### 5.2.3.8  setEndPointHostUsingName

Sets the value of the `Host` attribute of the specified end point for the specified server and node.

**Syntax**

```
OracleMWConfigUtilities.setEndPointHostUsingName(serverName,
nodeName,endPointName,hostValue)
```

| Argument | Definition |
|----------|------------|
| serverName | The name of the server |
| nodeName | The name of the node |
| endPointName | The name of the end point |
| hostValue | The host value, which may be `localhost`, the host name, or an IP address |

**Example**

The following example sets the value of the `Host` attribute of the `SoapListenerAddress` end point for the OracleAdminServer server on the TEST01Node03 node.

```
OracleMWConfigUtilities.setEndPointHostUsingName('OracleAdminServer','TEST01Node03
','SoapListenerAddress','myserver.domain.com')

Host Name myserver.domain.com for SoapListenerAddress is set.
```

### 5.2.3.9  setEndPointHostUsingObject

Sets the value of the `Host` attribute of the specified end point for the specified server.

**Syntax**

```
OracleMWConfigUtilities.setEndPointHostUsingObject(serverObject,
endPointName, hostValue)
```

| Argument | Definition |
|----------|------------|
| serverObject | The server object |
| endPointName | The name of the end point |
| hostValue | The host value, which may be `localhost`, the host name, or an IP address |

**Example**

The following example sets the value of the `Host` attribute of the `HttpListenerAddress` end point for the OracleAdminServer server.

```
server = OracleMWConfig.getChildByName('Server','OracleAdminServer')
OracleMWConfigUtilities.setEndPointHostUsingObject(server,'HttpListenerAddress',
'localhost')

Host Name localhost for HttpListenerAddress is set.
```

### 5.2.3.10 setEndPointPortUsingName

Sets the value of the `Port` attribute of the specified end point for the specified server name and node name.

**Syntax**

```
OracleMWConfigUtilities.setEndPointPortUsingName(serverName,
nodeName, endPointName, portValue)
```

| Argument | Definition |
|----------|------------|
| serverName | The name of the server |
| nodeName | The name of the node |
| endPointName | The name of the end point |
| portValue | The value to which to set the port |

**Example**

The following example sets the value of the `Port` attribute of the `HttpListenerAddress` end point for the OracleAdminServer server on the TEST01Node03 node.

```
OracleMWConfigUtilities.setEndPointPortUsingName('OracleAdminServer',
'TEST01Node03','HttpListenerAddress','9999')

Port 9999 for a HttpListenerAddress is set.
```

### 5.2.3.11 setEndPointPortUsingObject

Sets the value of the `Port` attribute of the specified end point for the specified server.

**Syntax**

```
OracleMWConfigUtilities.setEndPointPortUsingObject(serverObject,
endPointName, portValue)
```

| Argument | Definition |
|----------|------------|
| serverObject | The server object |
| endPointName | The name of the end point |
| portValue | The value to which to set the port |

**Example**

The following example sets the value of the `Port` attribute for the `HttpListenerAddress` end point for the OracleAdminServer server.

```
server = OracleMWConfig.getChildByName('Server','OracleAdminServer')
OracleMWConfigUtilities.setEndPointPortUsingObject(server,
'HttpListenerAddress','9999')

Port 9999 for a HttpListenerAddress is set.
```

## 5.2.4 JDBC Utility Functions

This section describes the following commands, which are used to manage JDBC data sources.

- getJdbcDatasourceNames
- getJdbcSchemaComponentNames
- showJdbc
- getJdbcURL
- getJdbcInstanceType
- getJdbcDriverClass
- validateJdbc
- createDatasource
- deleteDatasource
- setJdbcUsername
- getJdbcUsername
- setJdbcPassword
- setJdbcServerName
- getJdbcServerName
- setJdbcPortNumber
- getJdbcPortNumber
- setJdbcDatabaseName
- getJdbcDatabaseName
- setJdbcDriverVendor
- getJdbcDriverVendor
- setJdbcDriverXa
- getJdbcDriverXa
- setJdbcIsOracleInstanceType
- getJdbcIsOracleInstanceType

### 5.2.4.1 getJdbcDatasourceNames

Returns a list of normal JDBC data source instances.

**Syntax**

```
OracleMWConfigUtilities.getJdbcDatasourceNames()
```

**Example**

The following example indicates that there is only one normal JDBC data source in the cell configuration.

```
print OracleMWConfigUtilities.getJdbcDatasourceNames()

['TestNormalDatasource1']
```

### 5.2.4.2 getJdbcSchemaComponentNames

Returns a list of component schema instance names. It does not include any normal data source instance names.

**Syntax**

```
OracleMWConfigUtilities.getJdbcSchemaComponentNames()
```

**Example**

The following example indicates that there are four schemas in the cell configuration.

```
print OracleMWConfigUtilities.getJdbcSchemaComponentNames()

['SOA Infrastructure', 'User Messaging Service', 'SOA MDS Schema',
'OWSM MDS Schema']
```

### 5.2.4.3 showJdbc

Displays the list of configuration settings for the specified data source or component schema.

**Syntax**

```
OracleMWConfig.showJdbc(datasourceOrSchemaComponentName)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

The following example displays the list of configuration settings for the SOADatasource data source.

```
print OracleMWConfig.showJdbc('SOADatasource')

CFGFWK-68073:  The value of attribute Password is not allowed to be read.
[ar-] Name [SOADatasource]
[ar-] Type [Normal]
[arw] Username [websphere]
[a-w] Password [********]
[arw] DatabaseName [lnrac1]
[arw] ServerName [mydbserver.domain.com]
[arw] PortNumber [1521]
[ar-] URL [jdbc:oracle:thin:@mydbserver.domain.com:1521/lnrac1]

[arw] DatabaseType [Oracle]
[arw] DriverVendor [Oracle]
[arw] DriverType [4]
[arw] IsXa [true]
[arw] IsOracleInstanceType [false]
[arw] IsRac [false]
[ar-] DriverClass [oracle.jdbc.xa.client.OracleXADataSource]
```

### 5.2.4.4 getJdbcURL

Returns the value of the URL attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcURL(datasourceOrSchemaComponentNa
me)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

The following example returns the value of the URL attribute of the SOADatasource data source.

```
print OracleMWConfigUtilities.getJdbcURL('SOADatasource')
```

```
jdbc:oracle:thin:@mydbserver.domain.com:1521/lnrac1
```

### 5.2.4.5 getJdbcInstanceType

Returns the string Normal if the specified instance is a normal data source or returns Schema if the specified instance is a component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcInstanceType(datasourceOrSchemaCo
mponentName)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

The following example indicates that SOADatasource is a normal JDBC data source.

```
print OracleMWConfigUtilities.getJdbcInstanceType('SOADatasource')
```

```
Normal
```

### 5.2.4.6 getJdbcDriverClass

Returns the value of the DriverClass attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcDriverClass(datasourceOrSchemaCom
ponentName)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

The following example returns the value of the DriverClass attribute for the SOADatasource data source.

```
print OracleMWConfigUtilities.getJdbcDriverClass('SOADatasource')
```

```
oracle.jdbc.xa.client.OracleXADataSource
```

### 5.2.4.7 validateJdbc

Validates the configuration of the specified JDBC data source or component schema, and returns either `None` or an exception.

**Syntax**

```
OracleMWConfigUtilities.validateJdbc(datasourceOrSchemaComponent
Name)
```

| Argument | Definition |
| --- | --- |
| `datasourceOrSchemaComponentName` | The name of a data source or component schema |

**Examples**

The following example validates the configuration of the SOADatasource data source, and indicates that there are no validation errors.

```
print OracleMWConfigUtilities.validateJdbc('SOADatasource')

None
```

The following example validates the configuration of the SOADatasource data source, and indicates that a required parameter (`Host Name`) was not specified for the data source.

```
print OracleMWConfigUtilities.validateJdbc('SOADatasource')

CFGFWK-68065:  JDBC datasource validation failed: SOADatasource: CFGFWK-68021:
The following must be specified:
     Host Name
```

### 5.2.4.8 createDatasource

Creates a new data source using the specified attributes.

**Syntax**

```
OracleMWConfigUtilities.createDatasource(atts)
```

| Argument | Definition |
| --- | --- |
| `atts` | A list of attributes. See the example for the list of attributes that you can set for a data source. |

**Example**

The following example creates a data source named myNewDatasource with the specified attributes. In the example, *datasourceCustomKey* and *datasourceCustomValue* represents a custom name-value pair that may apply to the data source you are creating.

```
jdbcAtts = []
jdbcAtts.append(['Name','myNewDatasource'])
jdbcAtts.append(['Username','websphere'])
jdbcAtts.append(['Password','welcome1'])
jdbcAtts.append(['DatabaseType','Oracle'])
jdbcAtts.append(['DatabaseName','orcl'])
jdbcAtts.append(['ServerName','example.host.com'])
```

```
jdbcAtts.append(['PortNumber','1521'])
jdbcAtts.append(['IsXa','true'])
jdbcAtts.append(['StatementCacheSize','1024'])
jdbcAtts.append(['AgedTimeout','12345'])
jdbcAtts.append(['datasourceCustomKey','datasourceCustomValue'])
jdbcAtts.append(['JndiName','myJNDI'])
jdbcAtts.append(['Server','soa_server1'])
myNewDatasource = OracleMWConfigUtilities.createDatasource(jdbcAtts)
```

### 5.2.4.9 deleteDatasource

Deletes the specified data source or component schema.

#### Syntax

```
OracleMWConfigUtilities.deleteDatasource(datasourceOrSchmaCompon
entName)
```

| Argument | Definition |
|----------|------------|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

#### Example

```
OracleMWConfigUtilities.deleteDatasource('SOADatasource')
```

### 5.2.4.10 setJdbcUsername

Sets the value of the Username attribute of the specified data source or component schema.

#### Syntax

```
OracleMWConfigUtilities.setJdbcUsername(datasourceOrSchemaCompon
entName, username)
```

| Argument | Definition |
|----------|------------|
| datasourceOrSchemaComponentName | The name of a data source or component schema |
| username | The user name to use |

#### Example

This example sets the Username attribute for the SOADatasource data source to websphere.

```
OracleMWConfigUtilities.setJdbcUsername('SOADatasource', 'websphere')
```

### 5.2.4.11 getJdbcUsername

Returns the value of the Username attribute of the specified data source or component schema.

#### Syntax

```
OracleMWConfigUtilities.getJdbcUsername(datasourceOrSchemaCompon
entName)
```

| Argument | Definition |
| --- | --- |
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

This example returns the value of the Username attribute for the SOADatasource data source.

```
print OracleMWConfigUtilities.getJdbcUsername('SOADatasource')
```

```
websphere
```

### 5.2.4.12  setJdbcPassword

Sets the value of the Password attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcPassword(datasourceOrSchemaCompon
entName, password)
```

| Argument | Definition |
| --- | --- |
| datasourceOrSchemaComponentName | The name of a data source or component schema |
| password | The password to use |

**Example**

This example sets the password for the SOADatasource to welcome1.

```
OracleMWConfigUtilities.setJdbcPassword('SOADatasource', 'welcome1')
```

### 5.2.4.13  setJdbcServerName

Sets the value of the ServerName attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcServerName(datasourceOrSchemaComp
onentName, databaseServerName)
```

| Argument | Definition |
| --- | --- |
| datasourceOrSchemaComponentName | The name of a data source or component schema |
| databaseServerName | The name of the database server |

**Example**

This example sets the ServerName attribute for the SOADatasource data source to example.host.

```
OracleMWConfigUtilities.setJdbcServerName('SOADatasource', 'example.host')
```

### 5.2.4.14 getJdbcServerName

Returns the value of the `ServerName` attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcServerName(datasourceOrSchemaComp
onentName)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

This example returns the value of the `ServerName` attribute for the SOADataSource data source.

```
print OracleMWConfigUtilities.getJdbcServerName('SOADataSource')
```

```
myserver.domain.com
```

### 5.2.4.15 setJdbcPortNumber

Sets the value of the `PortNumber` attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcPortNumber(datasourceOrSchemaComp
onentName, listenPort)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |
| listenPort | The listen port to use on the database server |

**Example**

This example sets the `PortNumber` attribute for the SOADatasource data source to `1521`.

```
OracleMWConfigUtilities.setJdbcPortNumber('SOADatasource', '1521')
```

### 5.2.4.16 getJdbcPortNumber

Returns the value of the `PortNumber` attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcPortNumber(datasourceOrSchemaComp
onentName)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

This example returns the value of the PortNumber attribute for the SOADataSource data source.

```
print OracleMWConfigUtilities.getJdbcPortNumber('SOADatasource')

1521
```

### 5.2.4.17 setJdbcDatabaseName

Sets the value of the DatabaseName attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcDatabaseName(datasourceOrSchemaCo
mponentName, databaseName)
```

| Argument | Definition |
|----------|------------|
| datasourceOrSchemaComponentName | The name of a data source or component schema |
| databaseName | The name of the database |

**Example**

This example sets the value of the DatabaseName attribute for the SOADataSource data source to orc1.

```
OracleMWConfigUtilities.setJdbcDatabaseName('SOADatasource', 'orcl')
```

### 5.2.4.18 getJdbcDatabaseName

Returns the value of the DatabaseName attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcDatabaseName(datasourceOrSchemaCo
mponentName)
```

| Argument | Definition |
|----------|------------|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

This example returns the value of the DatabaseName attribute for the SOADatasource data source.

```
print OracleMWConfigUtilities.getJdbcDatabaseName('SOADatasource')

orc1
```

### 5.2.4.19 setJdbcDriverVendor

Sets the value of the DatabaseType attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcDriverVendor(datasourceOrSchemaCo
mponentName, databaseVendor)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |
| databaseVendor | The database vendor name |

**Example**

This example sets the value of the `DatabaseType` attribute for the SOADatasource data source to `Oracle`.

```
OracleMWConfigUtilities.setJdbcDriverVendor('SOADatasource', 'Oracle')
```

### 5.2.4.20 getJdbcDriverVendor

Returns the value of the `DatabaseType` attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcDriverVendor(datasourceOrSchemaCo
mponentName)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |

**Example**

This example returns the value of the `DatabaseType` attribute for the SOADatasource data source.

```
print OracleMWConfigUtilities.getJdbcDriverVendor('SOADatasource')
```

```
Oracle
```

### 5.2.4.21 setJdbcDriverXa

Sets the Boolean value of the `IsXa` attribute of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcDriverXa(datasourceOrSchemaCompon
entName, trueOrFalse)
```

| Argument | Definition |
|---|---|
| datasourceOrSchemaComponentName | The name of a data source or component schema |
| trueOrFalse | Set to `true` to indicate that the data source or component schema is XA. Set to `false` to indicate that the data source or component schema is non-XA. |

**Example**

The following example sets the `IsXa` attribute for the SOADatasource data source to
`true`.

```
OracleMWConfigUtilities.setJdbcDriverXa('SOADatasource', 'true')
```

### 5.2.4.22 getJdbcDriverXa

Returns the Boolean value of the `IsXA` attribute of the specified data source or
component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcDriverXa(datasourceOrSchemaCompon
entName)
```

| Argument | Definition |
|----------|------------|
| `datasourceOrSchemaComponentName` | The name of a data source or component schema |

**Example**

This example indicates that the SOADatasource data source is an XA data source.

```
print OracleMWConfigUtilities.getJdbcDriverXa('SOADatasource')

true
```

### 5.2.4.23 setJdbcIsOracleInstanceType

Sets the Oracle database URL to be an instance type (true) or a service type (false) for
the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcIsOracleInstanceType(datasourceOr
SchemaComponentName, trueOrFalse)
```

| Argument | Definition |
|----------|------------|
| `datasourceOrSchemaComponentName` | The name of a data source or component schema |
| `trueOrFalse` | Set to `true` if the Oracle database URL is an instance type. Set to `false` if the Oracle database URL is a service type. |

**Example**

```
OracleMWConfigUtilities.setJdbcIsOracleInstanceType('SOADatasource', 'false')
```

### 5.2.4.24 getJdbcIsOracleInstanceType

Returns the Boolean value of the `IsOracleInstanceType` attribute of the specified
data source or component schema. See setJdbcIsOracleInstanceType for related
information.

**Syntax**

```
OracleMWConfigUtilities.getJdbcIsOracleInstanceType(datasourceOr
SchemaComponentName)
```

| Argument | Definition |
|----------|------------|
| `datasourceOrSchemaComponentName` | The name of a data source or component schema |

**Example**

The following example indicates that the `IsOracleInstanceType` attribute for the SOADatasource data source is set to false, and is therefore a service type.

```
print OracleMWConfigUtilities.getJdbcIsOracleInstanceType('SOADatasource')
```

```
false
```

## 5.2.5 JDBC Utility Functions for Oracle RAC

This section describes the JDBC utility functions that are related to Oracle RAC.

- setJdbcRacHostsPorts

- setJdbcRacServiceName

- getJdbcRacServiceName

- convertJdbcNormalDatasourceToRAC

- convertJdbcRACToNormalDatasource

### 5.2.5.1 setJdbcRacHostsPorts

Sets the Oracle RAC database hosts and ports of the specified Oracle RAC data source or Oracle RAC component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcRacHostsPorts(datasourceOrSchemaC
omponentName, hostPortsList)
```

| Argument | Definition |
|----------|------------|
| `datasourceOrSchemaComponentName` | The name of the Oracle RAC data source or Oracle RAC component schema for which to set the hosts and ports |
| `hostPortsList` | A list of hosts and ports in the format `hostname:port`, such as `myhost:33`, `myhost2:33`, `myhost3:33` |

**Example**

The following example sets the Oracle RAC database hosts and ports for the SOADatasource data source to `example.host1:1521` and `example.host2:1521`.

```
OracleMWConfigUtilities.setJdbcRacHostsPorts('SOADatasource',
['example.host1:1521', 'example.host2:1521'])
```

### 5.2.5.2 setJdbcRacServiceName

Sets the Oracle RAC database service name for the specified Oracle RAC data source or Oracle RAC component schema.

**Syntax**

```
OracleMWConfigUtilities.setJdbcRacServiceName(datasourceOrSchema
ComponentName, racServiceName)
```

| Argument | Definition |
|----------|-----------|
| datasourceOrSchemaComponentName | The name of the Oracle RAC data source or Oracle RAC component schema for which to set the Service Name |
| racServiceName | The Oracle RAC Service Name to use |

**Example**

The following example sets the Oracle RAC database service name for the SOADatasource data source to `orc1`.

```
OracleMWConfigUtilities.setJdbcRacServiceName('SOADatasource', 'orcl')
```

### 5.2.5.3 getJdbcRacServiceName

Returns the Oracle RAC Service Name of the specified data source or component schema.

**Syntax**

```
OracleMWConfigUtilities.getJdbcRacServiceName(datasourceOrSchema
ComponentName)
```

| Argument | Definition |
|----------|-----------|
| datasourceOrSchemaComponentName | The name of the Oracle RAC data source or Oracle RAC component schema from which to get the Service Name |

**Example**

The following example returns the Oracle RAC service name for the SOADatasource data source.

```
OracleMWConfigUtilities.getJdbcRacServiceName('SOADatasource')
```

### 5.2.5.4 convertJdbcNormalDatasourceToRAC

Converts a normal data source or component schema to an Oracle RAC data source or schema with the specified data source name or component schema name, Oracle RAC service name, and a list of hosts and ports.

**Syntax**

```
OracleMWConfigUtilities.convertJdbcNormalDatasourceToRAC(datasou
rceName, RACserviceName, hostsPortsList)
```

| Argument | Definition |
|----------|-----------|
| datasourceOrSchemaComponentName | The data source or component schema name |
| RACserviceName | The Oracle RAC service name |
| hostPortsList | A list of hostname:port pairs |

**Example**

The following example converts the myDatasource data source from a normal JDBC data source to an Oracle RAC data source, sets the Service Name to `myRACService`, and sets the Oracle RAC database hosts and ports to `hostRac1.example.com:1521`, `hostRac2.example.com:1234`, and `hostRac3.example.com:4567`.

```
hostsPorts = ['hostRac1.example.com:1521',
'hostRac2.example.com:1234', 'hostRac3.example.com:4567']
OracleMWConfigUtilities.convertJdbcNormalDatasourceToRAC('myDatasource',
'myRACService', hostsPorts)
```

### 5.2.5.5 convertJdbcRACToNormalDatasource

Converts an Oracle RAC data source to a normal data source or component schema with the specified data source name or component schema name, database type, isXA flag, and database host and port.

**Syntax**

```
OracleMWConfigUtilities.convertJdbcRACToNormalDatasource(datasou
rceName, databaseType, isXA, dbHost, dbPort)
```

| Argument | Definition |
| --- | --- |
| datasourceName | The Oracle RAC data source name |
| databaseType | The database type |
| isXA | (Boolean) Set to `true` for XA; set to `false` for non-XA |
| dbHost | The database host name or IP address |
| dbPort | The database port |

**Example**

The following example converts the myDatasource data source from an Oracle RAC data source to a normal JDBC data source, using the specified data source values.

```
OracleMWConfigUtilities.convertJdbcRACToNormalDatasource('myDatasource',
'MSSQL Server', 'false', 'example.com', '1234')
```

## 5.2.6 File Store Utility Functions

This section describes the following utility functions for managing shared file stores.

- setFileStoreDirectory

- setLogDirectory

- setPermanentStoreDirectory

- setTemporaryStoreDirectory

- getFileStoreDirectory

- getLogDirectory

- getPermanentStoreDirectory

- getTemporaryStoreDirectory

> **Note:** You can use the following wsadmin command to display a list of service bus names:
>
> ```
> OracleMWConfig.getChildren('SIBus')
> ```

### 5.2.6.1 setFileStoreDirectory

Sets the sharable file store directory for the specified cluster name and bus name combination.

**Syntax**

```
OracleMWConfigUtilities.setFileStoreDirectory(type,clusterName,busName,directory)
```

| Argument | Definition |
|---|---|
| type | The type of directory, which can be LogDirectory, TemporaryStoreDirectory, or PermanentStoreDirectory |
| clusterName | The name of the cluster |
| busName | The name of the service bus |
| directory | The location that you want to use for the specified directory type. You must specify an absolute path. |

**Example**

The following examples sets the log directory for the sibus service bus on the cluster1 cluster.

```
OracleMWConfigUtilities.setFileStoreDirectory('LogDirectory','cluster1',
'sibus','${USER_INSTALL_ROOT}/log')
```

### 5.2.6.2 setLogDirectory

Sets the sharable log directory for the specified cluster name and bus name.

**Syntax**

```
OracleMWConfigUtilities.setLogDirectory(clusterName,busName,directory)
```

| Argument | Definition |
|---|---|
| clusterName | The name of the cluster |
| busName | The name of the bus |
| directory | The location that you want to use for the log directory. You must specify an absolute path. |

**Example**

The following example sets the log directory for the sibus service bus on the cluster1 cluster.

```
OracleMWConfigUtilities.setLogDirectory('cluster1','sibus'
'${USER_INSTALL_ROOT}/log')
```

### 5.2.6.3 setPermanentStoreDirectory

Sets the sharable permanent store directory for the specified cluster name and bus name.

**Syntax**

```
OracleMWConfigUtilities.setPermanentStoreDirectory(clusterName,b
usName,directory)
```

| Argument | Definition |
|---|---|
| clusterName | The name of the cluster |
| busName | The name of the bus |
| directory | The location that you want to use for the permanent store directory. You must specify an absolute path. |

**Example**

The following example sets the permanent store directory for the sibus service bus on the cluster1 cluster.

```
OracleMWConfigUtilities.setPermanentStoreDirectory('cluster1','sibus','${USER_
INSTALL_ROOT}/log')
```

### 5.2.6.4 setTemporaryStoreDirectory

Sets the sharable temporary store directory for the specified cluster name and bus name.

**Syntax**

```
OracleMWConfigUtilities.setTemporaryStoreDirectory(clusterName,b
usName,directory)
```

| Argument | Definition |
|---|---|
| clusterName | The name of the cluster |
| busName | The name of the bus |
| directory | The location that you want to use for the temporary store directory. You must specify an absolute path. |

**Example**

The following example sets the temporary store directory for the sibus service bus on the cluster1 cluster.

```
OracleMWConfigUtilities.setTemporaryStoreDirectory('cluster1','sibus','${USER_
INSTALL_ROOT}/log')
```

### 5.2.6.5 getFileStoreDirectory

Returns the sharable file store directory for a specified cluster name and bus name.

**Syntax**

```
OracleMWConfigUtilities.getFileStoreDirectory(type,
clusterName,busName)
```

| Argument | Definition |
| --- | --- |
| type | The type of directory to return |
| busName | The name of the bus |
| clusterName | The name of the cluster |

**Example**

The following example returns the log directory for the sibus service bus on the cluster1 cluster.

```
OracleMWConfigUtilities.getFileStoreDirectory('LogDirectory',
'cluster1','sibus')
```

```
${USER_INSTALL_ROOT}/log
```

### 5.2.6.6 getLogDirectory

Returns the sharable log directory of the specified cluster name and bus name.

**Syntax**

```
OracleMWConfigUtilities.getLogDirectory(clusterName,busName)
```

| Argument | Definition |
| --- | --- |
| clusterName | The name of the cluster |
| busName | The name of the bus |

**Example**

The following example returns the log directory for the sibus service bus on the cluster1 cluster.

```
OracleMWConfigUtilities.getLogDirectory('cluster1','sibus')
```

### 5.2.6.7 getPermanentStoreDirectory

Returns the sharable permanent store directory of the specified cluster name and bus name.

**Syntax**

```
OracleMWConfigUtilities.getPermanentStoreDirectory(clusterName,b
usName)
```

| Argument | Definition |
| --- | --- |
| clusterName | The name of the cluster |
| busName | The name of the bus |

**Example**

The following example returns the permanent store directory for the sibus service bus on the cluster1 cluster.

```
OracleMWConfigUtilities.getPermanentStoreDirectory('cluster1','sibus')
```

### 5.2.6.8 getTemporaryStoreDirectory

Returns the sharable temporary store directory of the specified cluster name and bus name.

**Syntax**

```
OracleMWConfigUtilities.getTemporaryStoreDirectory(clusterName,b
usName)
```

| Argument | Definition |
| --- | --- |
| cluster | The name of the cluster |
| busName | The name of the bus |

**Example**

The following example returns the temporary store directory for the sibus service bus on the cluster1 cluster.

```
OracleMWConfigUtilities.getTemporaryStoreDirectory('cluster1','sibus')
```

# Index

## V

## W