Oracle® Fusion Middleware

Third-Party Application Server Guide

11*g* Release 1 (11.1.1.6)

E17852-02

November 2011

ORACLE®

Oracle Fusion Middleware Third-Party Application Server Guide, 11*g* Release 1 (11.1.1.6)

E17852-02

Primary Author: Peter LaQuerre

Contributing Authors: Barbara Buerkle, Gail Flanegin, Helen Grembowicz, Peter Jew, Mark Kennedy, Liz Lynch, Robert May, Carlos Subi, Len Turmel

Contributors: Mike Blevins, Robert Campbell, Dave Felts, Jeni Ferns, Nick Greenhalgh, Harry Hsu, Hareesh Kolpuru, Vasant Kumar, Dennis Leung, Dan MacKinnon, Mark Miller, Kevin Minder, Vinod Nimmagadda, Vijay Ramanathan, Michael Rubino, Roy Sandjaja, Reza Shafii, Vishal Sharma, Stephen Sherman, Payal Srivastra, Sitaraman Swaminathan, Ken Vincent, Prakash Yamuna, Lisa Zitek-Jones

Contents

3 Managing Oracle Fusion Middleware on IBM WebSphere

4 Managing Oracle SOA Suite on IBM WebSphere

5 Managing Oracle WebCenter Portal on IBM WebSphere

6 Managing Oracle WebCenter Content on IBM WebSphere

7 Managing Web Services on IBM WebSphere

8 Managing Oracle Fusion Middleware Security on IBM WebSphere

9 Managing OAM Identity Assertion on IBM WebSphere

A Fusion Middleware Control Page Reference

Preface

This preface contains the following sections:

Audience

This manual is intended for Oracle Fusion Middleware system administrators who are responsible for installing and managing Oracle Fusion Middleware on third-party application servers, such as IBM WebSphere.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following related documentation available in the Oracle Fusion Middleware 11*g* documentation library:

- *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Application Development Framework*

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

Introduction to Third-Party Application Servers

This chapter introduces the Oracle Fusion Middleware 11*g* support for third-party application servers.

This chapter contains the following sections:

1.1 What Is a Third-Party Application Server?

A third-party application server is an application server provided by a vendor other than Oracle.

Oracle supports Oracle WebLogic Server as the primary platform for Oracle Fusion Middleware software components. However, to accommodate customers who want to run specific Oracle Fusion Middleware component software, such as Oracle SOA Suite, on application servers other than Oracle WebLogic Server, Oracle supports the third-party application servers described in this document.

1.2 Oracle Fusion Middleware Components That Support Third-Party Application Servers

You can configure the following Oracle Fusion Middleware components on supported third-party application servers:

- Oracle SOA Suite
- Oracle WebCenter Portal
- Oracle WebCenter Content
- Oracle Application Development Framework (Oracle ADF)
- Oracle Application Developer Runtime

For this release of Oracle Fusion Middleware 11*g*, Oracle supports only IBM WebSphere Application Server as a third-party application server for these Oracle Fusion Middleware products.

1.3 Overview of the Oracle Fusion Middleware IBM WebSphere Support

The following sections provide more detail about the supported Oracle Fusion Middleware features on IBM WebSphere:

- Supported IBM WebSphere Application Servers
- Understanding the Topology of Oracle Fusion Middleware on IBM WebSphere

1.3.1 Supported IBM WebSphere Application Servers

Oracle supports the following third-party application server products for specific Oracle Fusion Middleware products and certain Oracle Fusion Middleware configurations:

- IBM WebSphere Application Server - Network Deployment (ND) 7.0.0.17
- IBM WebSphere Application Server 7.0.0.17

Note that this information was valid at the time this document was published. For the most accurate and up-to-date information about the IBM WebSphere Application Server supported by Oracle Fusion Middleware, see the Certification information on the Oracle Technology Network (OTN), as described in Section 2.1, "Task 1: Review the System Requirements and Certification Information".

Note: In this guide, IBM WebSphere is used to reference both IBM WebSphere Application Server (AS) and IBM WebSphere Application Server Network Deployment (ND). The specific product names are used when appropriate.

1.3.2 Understanding the Topology of Oracle Fusion Middleware on IBM WebSphere

When you install and configure Oracle Fusion Middleware on IBM WebSphere, the resulting topology depends on whether you are running IBM WebSphere Application Server or IBM WebSphere Application Server - ND.

- Typical Oracle Fusion Middleware Topology on IBM WebSphere Application Server - ND
- Typical Oracle Fusion Middleware Topology on IBM WebSphere Application Server

1.3.2.1 Typical Oracle Fusion Middleware Topology on IBM WebSphere Application Server - ND

When you install and configure Oracle Fusion Middleware with IBM WebSphere Application Server - ND, the configuration process automatically creates an IBM WebSphere cell that contains a special server, in addition to the Deployment Manager, called the OracleAdminServer.

This OracleAdminServer hosts the key infrastructure pieces of Oracle Fusion Middleware, including the Java Required Files (JRF) and Oracle Enterprise Manager product templates:

- The JRF template provides important Oracle libraries and other capabilities that support new versions of APIs that many Oracle Fusion Middleware products and applications depend upon.

- The Oracle Enterprise Manager template provides Oracle Enterprise Manager Fusion Middleware Control, which you can use to manage the Oracle Fusion Middleware products you install and configure.

Additional products are installed on additional servers in the newly created IBM WebSphere cell.

When you configure your IBM WebSphere cell for use with Oracle Fusion Middleware, you can also include additional servers and clusters in your cell, and you can configure the Oracle Fusion Middleware products to work with an Oracle Real Application Clusters (Oracle RAC) database.

1.3.2.2 Typical Oracle Fusion Middleware Topology on IBM WebSphere Application Server

When you install and configure Oracle Fusion Middleware with IBM WebSphere Application Server, only one server is created. This one server is used both for administration and for application hosting.

1.4 Documentation Resources for Using Oracle Fusion Middleware on IBM WebSphere

You can refer to the following additional documentation resources for information about running Oracle Fusion Middleware on IBM WebSphere:

- The IBM WebSphere documentation available on the WebSphere Application Server Information Center for basic conceptual information about IBM WebSphere, as well details about installing IBM WebSphere.
- This document for an overview of the Oracle Fusion Middleware support for IBM WebSphere, a summary of the overall steps required to install and configure Oracle Fusion Middleware on IBM WebSphere, and a high-level listing of the features and tools available for installing and managing Oracle Fusion Middleware on IBM WebSphere.
- *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server* for complete information on the capabilities of the Oracle Fusion Middleware Configuration Wizard, including information about creating and modifying cells, how to add additional servers and clusters to a cell, and how to configure Oracle Fusion Middleware products to support an Oracle Real Application Clusters (Oracle RAC) database.
- Specific sections of the Oracle Fusion Middleware documentation library for information about specific feature areas described in this guide. As you review this document, note the links to specific Oracle documentation that can help you successfully develop and administer your Oracle Fusion Middleware applications on IBM WebSphere.

2

Installing and Configuring Oracle Fusion Middleware on IBM WebSphere

The following sections describe how to install and configure Oracle Fusion Middleware with IBM WebSphere.

> **Note:** this chapter provides basic information about how to install and configure a single instance of Oracle Fusion Middleware on IBM WebSphere. If you are interested in configuring a high availability environment on IBM WebSphere, then review the content in this chapter, and then see Section 3.4, "Configuring Oracle Fusion Middleware High Availability on IBM WebSphere".

2.1 Task 1: Review the System Requirements and Certification Information

Before performing any upgrade or installation you should read the system requirements documentation to ensure that your environment meets the minimum installation requirements for the products you are installing.

The system requirements document covers information such as hardware and software requirements, minimum disk space and memory requirements, and required system libraries, packages, or patches:

```
http://www.oracle.com/technology/software/products/ias/files/fusion_
requirements.htm
```

In addition, you should read the certification document. The certification document covers supported installation types, platforms, operating systems, databases, JDKs, and third-party products:

```
http://www.oracle.com/technology/software/products/ias/files/fusion_
certification.html
```

2.2 Task 2: Obtain the Necessary Software Media or Downloads

For this installation and configuration procedure, you will need to obtain the following software:

- IBM WebSphere 7.0 and any required Fix Packs for the IBM WebSphere software.

  At the time this document was published, the latest Fix Pack was Fix Pack 13 (7.0.0.13). For more information, see Section 2.4.1, "IBM Online Resources for Obtaining and Installing the IBM WebSphere Software.".

  For specific information the software requirements, refer to Section 2.1, "Task 1: Review the System Requirements and Certification Information".

- Oracle Fusion Middleware Repository Creation Utility 11*g* Release 1 (11.1.1.6.0) or later
- One of the following Oracle Fusion Middleware software products, which are supported on IBM WebSphere:
  - Oracle Application Development Runtime 11*g* Release 1 (11.1.1.6.0) or later
  - Oracle SOA Suite 11*g* Release 1 (11.1.1.6.0) or later
  - Oracle WebCenter Portal 11*g* Release 1 (11.1.1.6.0) or later
  - Oracle WebCenter Content 11*g* Release 1 (11.1.1.6.0)

  Note: The version numbers included here were accurate at the time this document was published. For specific software requirements, refer to the references in Section 2.1, "Task 1: Review the System Requirements and Certification Information".

  For information about where to download the software, refer to the *Oracle Fusion Middleware Download, Installation, and Configuration Readme Files* on the Oracle Technology Network (OTN):

  ```
  http://download.oracle.com/docs/cd/E23104_01/download_readme.htm
  ```

2.3 Task 3: Identify a Database and Install the Required Database Schemas

The following Oracle Fusion Middleware products require a metadata repository with required schemas to be installed in a supported database:

- Oracle SOA Suite
- Oracle WebCenter Portal
- Oracle WebCenter Content

You cannot configure these products without first installing the required schemas in a supported database.

To create or update schemas in a database, use the Repository Creation Utility (RCU).

> **Note:** It is recommended that all metadata repositories reside on a database at the same site as the products to minimize network latency issues.

For information about identifying the schemas required for specific Oracle Fusion Middleware products, as well as information about the database requirements and running RCU, refer to *Oracle Fusion Middleware Repository Creation Utility User's Guide*.

If you are installing Oracle WebCenter Content, then also refer to "Creating Oracle WebCenter Content Schemas with the Repository Creation Utility" in the *Oracle WebCenter Content Installation Guide*.

For information on the databases supported by Oracle Fusion Middleware, see the certification information described in Section 2.1, "Task 1: Review the System Requirements and Certification Information".

Make a note of the database connection information and the passwords for the schemas you create with the Repository Creation Utility. You will need these later when you configure the Oracle Fusion Middleware products.

2.4 Task 4: Install the IBM WebSphere Software

Oracle Fusion Middleware supports both the IBM WebSphere Application Developer - Network Deployment (ND) and IBM WebSphere Application Server (AS) products.

To install and configure Oracle Fusion Middleware with IBM WebSphere, you must first install (but not configure) IBM WebSphere 7.0 and apply the latest Fix Pack for IBM WebSphere 7.0.

Refer to the following sections for more information:

- IBM Online Resources for Obtaining and Installing the IBM WebSphere Software
- Important Considerations Before Installing the IBM WebSphere Software

2.4.1 IBM Online Resources for Obtaining and Installing the IBM WebSphere Software

Refer to the following IBM resources for more information.

Note that Oracle is not responsible for the content in the following links. These references are provided for convenience only. Be sure to refer to the IBM documentation provided with or referenced by your IBM WebSphere software distribution:

- To obtain and install the IBM WebSphere software, refer to the IBM WebSphere documentation. For more information, see Section 1.4, "Documentation Resources for Using Oracle Fusion Middleware on IBM WebSphere".
- For more information about the Fix Packs available for IBM WebSphere 7.0, refer to the Fix list for IBM WebSphere Application Server V7.0 on the IBM Support Web site.
- You install the Fix Packs using the IBM WebSphere Update Installer. For more information, see the information about the Maintenance Download Wizard for WebSphere Application Server V7.0 on the IBM Support Web site.

2.4.2 Important Considerations Before Installing the IBM WebSphere Software

Before you perform the IBM WebSphere installation, note the following requirements for Oracle Fusion Middleware products:

- Using the Correct IBM WebSphere Installer for Your Platform
- About the Sample Applications and Default Profiles During the IBM WebSphere Installation
- About the WAS_HOME Directory Path

2.4.2.1 Using the Correct IBM WebSphere Installer for Your Platform

Note that like Oracle WebLogic Server, IBM WebSphere is available for different platforms. Some platforms, such as Linux 64-bit platforms, require unique IBM WebSphere installers.

Before you begin your IBM WebSphere installation, be sure you have obtained the correct IBM WebSphere installer for your platform.

2.4.2.2 About the Sample Applications and Default Profiles During the IBM WebSphere Installation

Do not install any sample applications or create any profiles during the IBM WebSphere installation process.

The goal is to install the IBM WebSphere software on disk in a directory available to the Oracle Fusion Middleware software installation, which you will perform later. You will use the Oracle Fusion Middleware Configuration wizard to configure the required IBM WebSphere profiles.

2.4.2.3 About the WAS_HOME Directory Path

When you install the IBM WebSphere software, you are prompted for the location where you want to install the software. For the purposes of this documentation, this location is later referred to as the WAS Home, or WAS_HOME in examples.

If you accept the default values that are provided during the installation, then the WAS_HOME is installed in the following directory structure:

```
DISK/IBM/WebSphere/Application Server
```

Create the WAS_HOME for the IBM WebSphere software on the same host where you plan to install the Oracle Fusion Middleware software.

Make a note of this path. You will be asked to identify the location of the IBM WebSphere directory when you configure Oracle Fusion Middleware.

2.5 Task 5: Install Oracle Fusion Middleware

The following sections provide information on installing Oracle Fusion Middleware with IBM WebSphere:

- General Installation Instructions for the Supported Oracle Fusion Middleware Products
- Special Instructions When Installing Oracle Fusion Middleware with IBM WebSphere

2.5.1 General Installation Instructions for the Supported Oracle Fusion Middleware Products

For general instructions on installing any of the Oracle Fusion Middleware products that are supported on IBM WebSphere, refer to Table 2–1.

Table 2–1 Locating Installation Information for Oracle Fusion Middleware Products

Product	Installation Instructions
Oracle Application Developer Runtime	"Installation Instructions" in the *Oracle Fusion Middleware Installation Guide for Application Developer*
Oracle SOA Suite	"Installation Instructions" in the *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite*
Oracle WebCenter Portal	"Installing Oracle WebCenter Portal" in the *Oracle Fusion Middleware Installation Guide for Oracle WebCenter Portal*
Oracle WebCenter Content	"Using the Installer for Oracle WebCenter Content" in the *Oracle WebCenter Content Installation Guide*

2.5.2 Special Instructions When Installing Oracle Fusion Middleware with IBM WebSphere

Note the following special instructions that apply when you are installing Oracle Fusion Middleware products on IBM WebSphere:

- When you are prompted to specify a JRE/JDK location, you can specify the following directory in the IBM WebSphere home:

  ```
  (UNIX) WAS_HOME/java
  (Windows) WAS_HOME\java
  ```

  For example, if you are using the default location for a typical IBM WebSphere Application Server directory on a UNIX operating system:

  ```
  diskname/IBM/WebSphere/AppServer/java
  ```

- Before installing Oracle WebCenter Content, you need to change the Java socket factories to the default JSSE implementation. For more information, see Section 6.1.1, "Changing Java Socket Factories in the IBM JDK."
- When you are prompted to provide a Middleware home, note that you can enter a new Middleware home directory path.

  When you install Oracle Fusion Middleware products on Oracle WebLogic Server, you create the Middleware home, when you install Oracle WebLogic Server. This is because Oracle WebLogic Server is included in the Middleware home.

  In contrast, when you install Oracle Fusion Middleware on IBM WebSphere, you create the Middleware home when you install the Oracle Fusion Middleware software. This is because the IBM WebSphere software is not installed inside the Middleware home. It is installed in a separate directory structure.
- When you select IBM WebSphere as your application server and you are prompted for the Application Server Location, enter the path to the IBM WebSphere application server directory you created in Section 2.4, "Task 4: Install the IBM WebSphere Software".

  For example:

  ```
  diskname/IBM/WebSphere/AppServer/
  ```

2.6 Task 6: Configure an LDAP Server

Most Oracle Fusion Middleware components require a supported LDAP server. However, an LDAP server is not automatically installed and configured when you install Oracle Fusion Middleware components on IBM WebSphere. Oracle Fusion Middleware components do not support WebSphere's built-in file-based user registry

Before you can configure Oracle SOA Suite, Oracle WebCenter Portal, Oracle WebCenter Content, or Oracle Fusion Middleware Web services in a new IBM WebSphere cell, you must install and configure a supported external LDAP server, such as Oracle Internet Directory.

2.6.1 General Information About Supported LDAP Servers and Identity Stores

For information about the LDAP servers that Oracle Fusion Middleware supports, see the certification information on the Oracle Technology Network (OTN):

`http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html`

For information about installing a supported LDAP server, see Section 8.1, "IBM WebSphere Identity Stores".

2.6.2 Oracle Fusion Middleware Component-Specific LDAP Information

For information about configuring users and roles for each Oracle Fusion Middleware component that is supported on IBM WebSphere, see the appropriate section below:

- Section 4.1.1, "Configuring SOA Suite Users and Groups in an External LDAP Server".
- Section 5.2.7, "Configuring an Admin User for the Spaces Application"
- Section 5.2.8, "Configuring an Admin User for the Discussions Server"
- Section 6.2.3, "Configuring an LDAP Server for Oracle WebCenter Content Users and Groups on IBM WebSphere"
- Section 7.1, "Configuring a Default Administrative User from the LDAP Directory"

2.7 Task 7: Configure Your Oracle Fusion Middleware Components in a New IBM WebSphere Cell

To configure Oracle Fusion Middleware components in an IBM WebSphere environment, you use a special version of the Oracle Fusion Middleware Configuration Wizard.

This section describes how to use the Configuration Wizard to configure your Oracle Fusion Middleware products in a simple IBM WebSphere cell. For complete information about using the Oracle Fusion Middleware Configuration Wizard, including information about adding servers and clusters to a cell, refer to the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

Note: The instructions here describe how to use the Configuration Wizard to configure your components. However, you can also use the WebSphere wsadmin command-line utility to configure your Oracle Fusion Middleware components.

- For more information about using the wsadmin command-line utility, see Section 3.1.3, "Using the Oracle Fusion Middleware wsadmin Commands".
- For more information about configuring components with wsadmin, see "Using wsadmin to Configure Oracle Fusion Middleware" in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

To configure your Oracle Fusion Middleware product in a new IBM WebSphere cell:

1. If you have installed the Oracle Fusion Middleware schemas in an IBM DB2 database, then be sure to perform the required preconfiguration steps.

   For more information, see "Before You Begin" in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

2. Start the Oracle Fusion Middleware Configuration Wizard by running the following command in the Oracle home of the product you want to configure:

   ```
   (UNIX) MW_HOME/ORACLE_HOME/common/bin/was_config.sh
   (Windows) MW_HOME\ORACLE_HOME\common\bin\was_config.cmd
   ```

   Consider the following notes when starting the Configuration Wizard:

   - Be sure to start the IBM WebSphere version of the Configuration Wizard. For more information, see "Starting the Configuration Wizard" in *Oracle Fusion Middleware Creating WebSphere Cells Using the Configuration Wizard*.
   - In the preceding example, note that you must replace the *ORACLE_HOME* with the path to the Oracle home of the product you are about to configure. For example, if you are configuring an Oracle SOA home, enter the following on a UNIX system:

     ```
     SOA_ORACLE_HOME/common/bin/was_config.sh
     ```

3. Follow the instructions on the screen to configure a new IBM WebSphere cell.

   Refer to the following for more information:

   - General Information About Using the Configuration Wizard on IBM WebSphere
   - Component-Specific Information About Using the Configuration Wizard on IBM WebSphere

2.7.1 General Information About Using the Configuration Wizard on IBM WebSphere

Note the following information as you advance through the Configuration Wizard:

- Be sure to make a note of the values you enter on the Specify Cell, Profile, and Node Name Information screen. You will need these later when you are starting and managing the cell. In particular, make note of the values you enter in the **Deployment Manager Profile Name** field and the **Application Server Profile Name** field.

- When the Add Products to Cell screen appears, refer to the following:

  "Fusion Middleware Product Templates" in the *Oracle Fusion Middleware Domain Template Reference* if you have questions about what capabilities are configured when you select each template.

  Component-Specific configuration information in the appropriate chapter of this guide. For more information, see Section 2.7.2, "Component-Specific Information About Using the Configuration Wizard on IBM WebSphere".

- If you select a product that requires a database schema, you will be prompted for database connection information for each required schema. To fill out this screen, use the database and schema information you noted in Section 2.3, "Task 3: Identify a Database and Install the Required Database Schemas".

- When you are prompted for advanced options, you can click **Next** and use the default settings. Refer to Section 1.3.2, "Understanding the Topology of Oracle Fusion Middleware on IBM WebSphere" for information on the topologies that will be created using the default settings.

  If you wish to modify the default settings (for example, if you want to target the products to different servers in the cell), refer to *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

2.7.2 Component-Specific Information About Using the Configuration Wizard on IBM WebSphere

For component-specific configuration information, refer to the following:

- Chapter 4, "Managing Oracle SOA Suite on IBM WebSphere"
- Chapter 5, "Managing Oracle WebCenter Portal on IBM WebSphere"
- Chapter 6, "Managing Oracle WebCenter Content on IBM WebSphere"

2.8 Task 8: Start the IBM WebSphere Servers

After you finish configuring the Oracle Fusion Middleware software successfully, you can start the IBM WebSphere Deployment Manager, Node, and Servers.

The following procedure shows the sequence you must use to start the deployment manager, the node, and the servers in the cell.

> **Note:** Before you start any Oracle WebCenter Content Node or Server, see "Verifying the Oracle WebCenter Content Configuration" in the *Oracle WebCenter Content Installation Guide* for information about postinstallation tasks that need to be completed before you log in to a server for the first time.

In the following examples, replace the names of the deployment manager and profile name with the values you entered in the Configuration Wizard in Section 2.7, "Task 7: Configure Your Oracle Fusion Middleware Components in a New IBM WebSphere Cell":

1. Start the Deployment Manager:

   Navigate to the following directory in the IBM WebSphere home and enter the following command:

```
(UNIX) profiles/deployment_mgr_name/bin/startManager.sh
                -profileName dmgr_profileName
(Windows) profiles\deployment_mgr_name\bin\startManager.cmd
                -profileName dmgr_profileName
```

For example, on a UNIX operating system:

```
/disk01/IBM/WebSphere/AppServer/profiles
       /Dmgr01/bin/startManager.sh -profileName Dmgr01
```

2. Start the node:

   Navigate to the following directory in the IBM WebSphere home and enter the following command:

   ```
   (UNIX) profiles/profile_name/bin/startNode.sh -profileName profileName
   (Windows) profiles\profile_name\bin\startNode.cmd -profileName profileName
   ```

   For example, on a UNIX operating system:

   ```
   /disk01/IBM/WebSphere/AppServer/profiles /Custom01/bin/startNode.sh
                   -profileName custom01
   ```

3. Start the OracleAdminServer server:

   Navigate to the following directory in the IBM WebSphere home and enter the following command:

   ```
   (UNIX) profiles/profile_name/bin/startServer.sh OracleAdminServer
           -profileName profileName
   (Windows) profiles\profile_name\bin\startServer.cmd OracleAdminServer
           -profileName profileName
   ```

   For example, on a UNIX operating system:

   ```
   /disk01/IBM/WebSphere/AppServer/profiles/Custom01/bin/startServer.sh
           OracleAdminSErver
           -profileName Custom01
   ```

4. Start any additional servers that were configured as part of your IBM WebSphere cell.

   After you start the OracleAdminServer, you can start the other servers using the IBM WebSphere Administrative Console or Oracle Enterprise Manager Fusion Middleware Control. For more information, see Section 3.1, "Summary of the Oracle Fusion Middleware Management Tools on IBM WebSphere".

   Alternatively, you can use the startServer script, as follows:

   Navigate to the following directory in the IBM WebSphere home and enter the following command:

   ```
   (UNIX) profiles/profile_name/bin/startServer.sh server_name
           -profileName profileName
   profiles\profile_name\bin\startServer.cmd server_name
           -profileName profileName
   ```

   For example, for an Oracle SOA Suite cell on a UNIX operating system:

   ```
   /disk01/IBM/WebSphere/AppServer/profiles
           /Custom01/bin/startServer.sh soa_server1
           -profileName Custom01
   ```

The typical servers that are configured for each of the Oracle Fusion Middleware components are listed in Table 2–2.

Table 2–2 Typical Oracle Fusion Middleware Component-Specific Managed Servers in an IBM WebSphere Cell

Component	Typical Managed Servers
Oracle SOA Suite	soa_server1, bam_server1
Oracle WebCenter Portal	WC_Spaces, WC_Collaboration, WC_Portlet, WC_Utilities
Oracle WebCenter Content	UCM_server1, URM_server1, or IBR_server1

2.9 Task 9: Verify the Configuration of the IBM WebSphere Cell

To verify the installation, use the IBM WebSphere Administration Console and Oracle Enterprise Manager Fusion Middleware Control to verify that the management tools are working and the servers are up and running.

Refer to Section 3.1, "Summary of the Oracle Fusion Middleware Management Tools on IBM WebSphere" for more information on locating the URLs for these Web-based management tools.

3

Managing Oracle Fusion Middleware on IBM WebSphere

This chapter provides basic information about managing Oracle Fusion Middleware on IBM WebSphere. This chapter contains the following topics:

- Summary of the Oracle Fusion Middleware Management Tools on IBM WebSphere
- Basic Administration Tasks on IBM WebSphere
- Deploying Applications on IBM WebSphere
- Configuring Oracle Fusion Middleware High Availability on IBM WebSphere

3.1 Summary of the Oracle Fusion Middleware Management Tools on IBM WebSphere

After you install and configure Oracle Fusion Middleware on IBM WebSphere, you can verify the configuration, and monitor and manage the components of the Oracle Fusion Middleware installation, using one of several management tools.

The following sections introduce the management tools:

- Using the WebSphere Administrative Console
- Using Oracle Enterprise Manager Fusion Middleware Control
- Using the Oracle Fusion Middleware wsadmin Commands

3.1.1 Using the WebSphere Administrative Console

This section contains the following topics:

- About the IBM WebSphere Administrative Console
- Locating the Port Number and URL of the IBM WebSphere Administrative Console

3.1.1.1 About the IBM WebSphere Administrative Console

The IBM WebSphere Administrative Console, also known as the IBM WebSphere Integrated Solutions Console, provides a Web-based interface for managing the IBM WebSphere environment.

Note that you cannot manage Oracle Fusion Middleware products, such as Oracle SOA Suite, Oracle WebCenter Portal, or Oracle WebCenter Content, from the IBM WebSphere Administrative Console, but you can use the console to monitor and

manage the cell and the servers on which the Oracle Fusion Middleware products are deployed.

For more information about the IBM WebSphere Administrative Console, see the IBM WebSphere documentation, as well as the online help for the console.

3.1.1.2 Locating the Port Number and URL of the IBM WebSphere Administrative Console

Before you can display the IBM WebSphere Administrative Console, you must identify the port number on which is running.

To locate the port number and URL of the IBM WebSphere Administrative Console:

1. In a text editor, open the following properties file:

   WAS_HOME/profiles/deployment_mgr_name/properties/portdef.props

2. Locate the value of the WC_Adminhost property.
3. Open a browser and enter the following URL:

   http://*hostname*:*WC_Adminhost_port*/ibm/console

   For example:

   http://host42.example.com:9002/ibm/console

3.1.2 Using Oracle Enterprise Manager Fusion Middleware Control

This section contains the following topics:

- About Oracle Enterprise Manager Fusion Middleware Control
- Locating the Port Number and URL for Fusion Middleware Control
- Displaying Fusion Middleware Control
- Viewing an IBM WebSphere Cell from Fusion Middleware Control
- Viewing an IBM WebSphere Server from Fusion Middleware Control
- Viewing an IBM WebSphere Application Deployment from Fusion Middleware Control
- Performing Oracle Fusion Middleware-Specific Administration Tasks for the Cell
- Differences When Using Fusion Middleware Control on IBM WebSphere

3.1.2.1 About Oracle Enterprise Manager Fusion Middleware Control

Oracle Enterprise Manager Fusion Middleware Control is a Web browser-based, graphical user interface that you can use to monitor and administer Oracle Fusion Middleware.

Fusion Middleware Control organizes a wide variety of performance data and administrative functions into distinct, Web-based home pages for cells, servers, components, and applications. The Fusion Middleware Control home pages make it easy to locate the most important monitoring data and the most commonly used administrative functions from your Web browser.

For more information, refer to "Getting Started Using Oracle Enterprise Manager Fusion Middleware Control" in the *Oracle Fusion Middleware Administrator's Guide*.

Note that the information provided in the *Oracle Fusion Middleware Administrator's Guide* is specific to using Fusion Middleware Control on Oracle WebLogic Server. For more information, see Section 3.1.2.8, "Differences When Using Fusion Middleware Control on IBM WebSphere".

3.1.2.2 Locating the Port Number and URL for Fusion Middleware Control

To locate the port number for Fusion Middleware Control:

1. Use your Web browser to open the IBM WebSphere Administrative Console.
2. In the navigation panel, select **Servers > Server Types > WebSphere application servers**.
3. Click **OracleAdminServer** to display the configuration properties of the server.
4. In the Communications section of the resulting page, expand **Ports** to list the important port values for the OracleAdminServer.
5. Locate the value of the `WC_Defaulthost` port.

3.1.2.3 Displaying Fusion Middleware Control

To display Fusion Middleware Control, create a new Web browser window or tab, and enter the following URL:

`http://`*`hostname`*`:`*`WC_Defaulthost_port`*`/em`

For example:

```
http://host42.example.com:9002/em
```

Log in to Fusion Middleware Control using the same administration credentials you use when logging in to the IBM WebSphere Administrative Console.

3.1.2.4 Viewing an IBM WebSphere Cell from Fusion Middleware Control

From Fusion Middleware Control, you can manage the Oracle Fusion Middleware products that you have installed and configured as part of the IBM WebSphere cell.

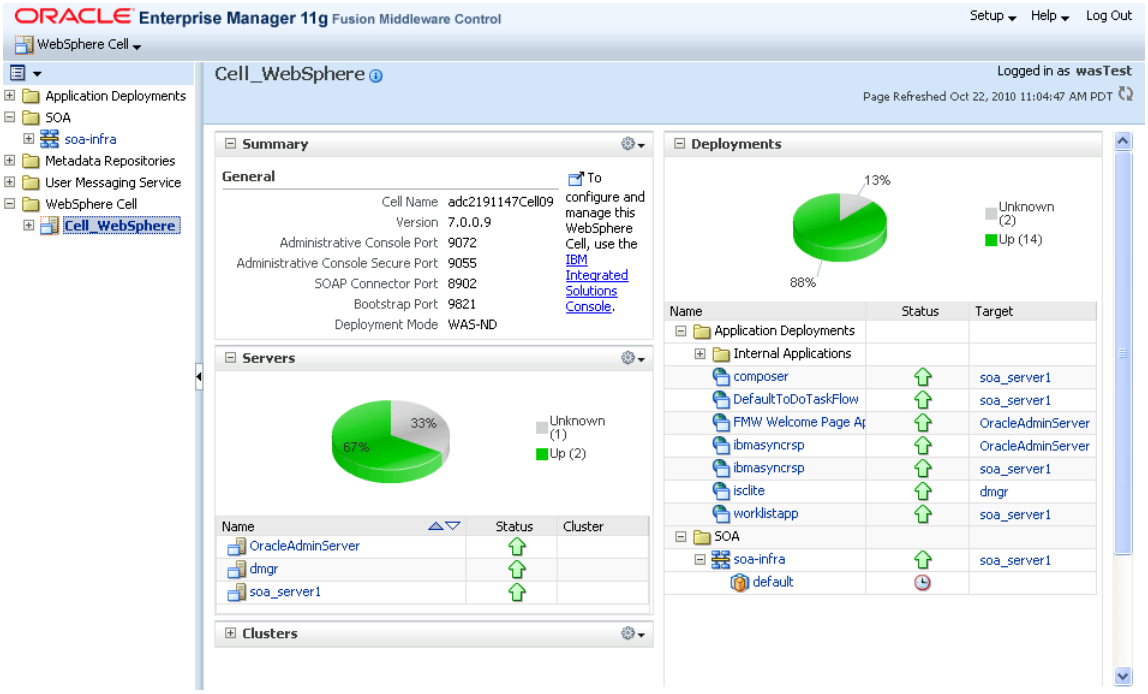
When you first log in to Fusion Middleware Control, the IBM WebSphere Cell home page appears (Figure 3–1). From this page, you can view the servers, applications, and clusters that are associated with the cell.

You can also navigate to the management pages for the Oracle Fusion Middleware components you have installed and configured. For example, if you have installed and configured Oracle SOA Suite, then expand the **SOA** folder in the Target Navigation Pane, and then click **soa-infra** to administer and monitor the SOA Infrastructure.

For more information about how to navigate within Oracle Enterprise Manager Fusion Middleware Control, see "Navigating Within Fusion Middleware Control" in the *Oracle Fusion Middleware Administrator's Guide*.

From the **WebSphere Cell** menu, you can perform Oracle Fusion Middleware administration functions. For help on a menu command, select the command, and then select **Enterprise Manager Help** from the **Help** menu on the resulting page.

Figure 3–1 Viewing the IBM WebSphere Cell from Fusion Middleware Control



3.1.2.5 Viewing an IBM WebSphere Server from Fusion Middleware Control

Each server in an IBM WebSphere cell has its own home page in Fusion Middleware Control.

To view the home page for a specific server:

1. In the Fusion Middleware Control Target Navigation Pane, expand the **WebSphere Cell** folder.
2. Expand the cell name, and click the server name.

From the WebSphere Application Server home page you can view general information about the server, display the IBM WebSphere Administrative Console, and view the status of the applications deployed to the server.

For a description of the features and options available on the IBM WebSphere Application Server home page, see Section A.1, "Understanding the Information on the IBM WebSphere Cell Home Page".

From the **WebSphere Application Server** menu, you can perform Oracle Fusion Middleware administration functions. For help on a menu command, select the command, and then--on the resulting page--select **Enterprise Manager Help** from the **Help** menu.

3.1.2.6 Viewing an IBM WebSphere Application Deployment from Fusion Middleware Control

Each application deployment in your IBM WebSphere cell has its own home page in Fusion Middleware Control.

An application deployment is an instance of a deployed application. For example, if you deploy the same application to two servers, then you have two deployments of the same application.

To view an application deployment in Fusion Middleware Control:

1. Navigate to the IBM WebSphere cell home page or an IBM WebSphere application server home page.
2. Locate the list of application deployments, and click the application name.

For a description of the features and options available on the IBM WebSphere Application Server home page, see Section A.3, "Understanding the Information on the IBM WebSphere Application Deployment Home Page".

From the **Application Deployment** menu, you can perform Oracle Fusion Middleware administration functions. For help on a menu command, select the command, and then--on the resulting page--select **Enterprise Manager Help** from the **Help** menu.

3.1.2.7 Performing Oracle Fusion Middleware-Specific Administration Tasks for the Cell

Oracle Enterprise Manager Fusion Middleware Control, when used with the IBM WebSphere Administrative Console, provides you with the tools you need to manage Oracle Fusion Middleware when it is installed and configured on IBM WebSphere.

You perform common IBM WebSphere administration tasks from the IBM WebSphere Administrative Console, and you can perform administration tasks that are specific to Oracle Fusion Middleware from the Fusion Middleware Control home pages.

3.1.2.8 Differences When Using Fusion Middleware Control on IBM WebSphere

When you use Oracle Enterprise Manager Fusion Middleware Control to manage Oracle Fusion Middleware products on IBM WebSphere, you will notice some differences from the features and functionality available when using it with Oracle WebLogic Server.

The differences vary, depending on whether you are using IBM WebSphere - Network Deployment (ND) or IBM WebSphere Application Server (AS).

Some specific menu commands and features available in an Oracle WebLogic Server environment are not available when you are managing Oracle Fusion Middleware in an IBM WebSphere environment. If a command or feature is not available, then it is not supported in the IBM WebSphere environment.

Table 3–1 describes some of the differences you might experience when managing Oracle Fusion Middleware on an IBM WebSphere cell, as opposed to an Oracle WebLogic Server domain.

Table 3–1 Summary of Differences When Managing IBM WebSphere As Opposed to Oracle WebLogic Server Domain

Feature or Functional Area	Differences on IBM WebSphere ND	Additional differences on IBM WebSphere AS
Managing an Oracle Fusion Middleware Farm	There is no concept of an Oracle Fusion Middleware farm when you are running on IBM WebSphere; instead, the first page that Fusion Middleware Control displays when you log in is the IBM WebSphere Cell home page. From the Cell home page, you can navigate to the other home pages that have monitoring and administrative features for the Oracle Fusion Middleware components. You can also link easily to the IBM WebSphere Administrative Console when necessary.	Same as ND.
Monitoring IBM WebSphere from Fusion Middleware Control	There are no IBM WebSphere performance metrics and no performance summary page for the IBM WebSphere cell or server pages.	Same as ND.
Deployment of Fusion Middleware Control in the cell	When you are managing an IBM WebSphere cell, Fusion Middleware Control runs on the OracleAdminServer, which is created when you configure Oracle Fusion Middleware products using the Configuration Wizard. You can then use Fusion Middleware Control to manage all the servers and applications deployed to the servers in the cell.	Single instance management only. Fusion Middleware Control must be running on the server that is being managed.
Application deployment from Fusion Middleware Control	You cannot deploy applications from Fusion Middleware Control on IBM WebSphere. Instead, you can use the IBM WebSphere Administrative Console or deploy directly from Oracle JDeveloper. For more information, see Section 3.3, "Deploying Applications on IBM WebSphere".	Same as ND.
Management of SOA Applications.	See Chapter 4, "Managing Oracle SOA Suite on IBM WebSphere".	See Chapter 4, "Managing Oracle SOA Suite on IBM WebSphere".
Management of Oracle WebCenter Portal Applications	See Chapter 5, "Managing Oracle WebCenter Portal on IBM WebSphere".	See Chapter 5, "Managing Oracle WebCenter Portal on IBM WebSphere"
Management of Oracle WebCenter Content	See Chapter 6, "Managing Oracle WebCenter Content on IBM WebSphere".	See Chapter 6, "Managing Oracle WebCenter Content on IBM WebSphere".
Management of Oracle Fusion Middleware Web services.	See Chapter 7, "Managing Web Services on IBM WebSphere".	See Chapter 7, "Managing Web Services on IBM WebSphere".

Table 3–1 (Cont.) Summary of Differences When Managing IBM WebSphere As Opposed to Oracle WebLogic Server Domain

Feature or Functional Area	Differences on IBM WebSphere ND	Additional differences on IBM WebSphere AS
Management of Oracle Platform Security Services (OPSS) features	See Chapter 8, "Managing Oracle Fusion Middleware Security on IBM WebSphere"	See Chapter 8, "Managing Oracle Fusion Middleware Security on IBM WebSphere"

3.1.3 Using the Oracle Fusion Middleware wsadmin Commands

The WebSphere Application Server wsadmin tool is a command-line utility that can be run in two modes:

- Interactive mode, where you enter commands directly in the shell
- Scripting mode, where you specify a Jython (.py) script on the command line

The examples in this chapter assume you are using interactive mode and the wsadmin command-line shell. For information about using scripting mode, refer to the IBM WebSphere documentation.

You can use the wsadmin tool to manage WebSphere Application Server as well as the configuration, application deployment, and server run-time operations.

Oracle Fusion Middleware provides a set of wsadmin commands that are used exclusively to manage the Oracle Fusion Middleware components that are configured in your IBM WebSphere cell.

For more information about the Oracle Fusion Middleware wsadmin commands and how to use them, refer to the following sections:

- Section 3.1.3.1, "About the Oracle Fusion Middleware wsadmin Command-Line Shell"
- Section 3.1.3.2, "Starting the Oracle Fusion Middleware wsadmin Command-Line Shell and Connecting to the Deployment Manager"
- Section 3.1.3.3, "Using the Oracle Fusion Middleware wsadmin Command-Line Online Help"
- Section 3.1.3.4, "Differences Between the wsadmin Commands and the WebLogic Scripting Tool (WLST) Commands"
- Section 3.1.3.5, "Differences Between Oracle Fusion Middleware wsadmin Commands and IBM WebSphere Wsadmin Commands"

3.1.3.1 About the Oracle Fusion Middleware wsadmin Command-Line Shell

A command-line shell is a command-line environment where a specific set of commands are available and supported. Within the shell, you can run these commands, obtain help on the commands, and perform administration tasks that are specific to the environment you are managing.

The Oracle Fusion Middleware wsadmin command-line shell is an Oracle Fusion Middleware-specific implementation of the wsadmin tool. From this shell, you can:

- Run the Oracle Fusion Middleware-specific wsadmin commands.
- List the available Oracle Fusion Middleware wsadmin commands.
- Obtain online help for the Oracle Fusion Middleware wsadmin commands.

3.1.3.2 Starting the Oracle Fusion Middleware wsadmin Command-Line Shell and Connecting to the Deployment Manager

Start the Oracle Fusion Middleware wsadmin command-line shell from `common/bin` directory of the Oracle home of the product you are managing.

For a complete list of the arguments you can use when starting wsadmin, refer to the IBM WebSphere documentation.

In a typical Oracle Fusion Middleware wsadmin session, you will want to specify the profile name and connect to the deployment manager of the cell you are managing.

> **Note:** The following examples assume you have already installed and configured an IBM WebSphere cell, using the instructions in Chapter 2, "Installing and Configuring Oracle Fusion Middleware on IBM WebSphere".
>
> Alternatively, if you want to run the wsadmin shell before configuring a cell, refer to "Prerequisite Environment Setup" in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

The following example shows how you can start the wsadmin shell.

Note that this example assumes the IBM WebSphere Deployment Manager is on the local host and is using the default SOAP port. If the Deployment Manager is on a different host, then you will need to specify the host and port using additional command-line arguments. For more information, see the IBM WebSphere documentation and wsadmin command-line help.

To start the wsadmin shell, use this command syntax:

```
(UNIX) ORACLE_HOME/common/bin/wsadmin.sh
      -profileName profilename
      -connType SOAP
      -user admin_user
      -password admin_password

(Windows) ORACLE_HOME\common\bin\wsadmin.cmd
     -profileName profilename
      -connType SOAP
      -user admin_user
      -password admin_password
```

The following example uses the complete path for the wsadmin script on a UNIX operating system:

```
/disk01/Oracle/Middleware/Oracle_SOA1/common/bin/wsadmin.sh -profileName soaDmgr05
```

Example 3–1 shows an example of starting the Oracle Fusion Middleware wsadmin command-line shell after you have changed directory to the `common/bin` directory in the Oracle Fusion Middleware product Oracle home on a UNIX system. The example also shows some typical output messages when you start the shell.

Example 3–1 Starting the Oracle Fusion Middleware Wsadmin Command-Line Shell

```
 ./wsadmin.sh -profileName soaDmgr05 -connType SOAP -user wasTest -password welcome1
IN SOA WsadminEnv.sh...
WSADMIN_CLASSPATH=:/scratch/wasTest/mwhome_soa_100719/oracle_common/soa/modules/oracle.soa.mgmt
_11.1.1/soa-infra-mgmt.jar:/scratch/wasTest/mwhome_soa_100719/ ...
    .
```

```
    .
    .
WASX7209I: Connected to process "dmgr" on node soaCellManager05 using SOAP connector;  The type of
process is: DeploymentManager

CFGFWK-24021:  OracleHelp loaded.
CFGFWK-24022:  For information on Oracle modules enter 'print OracleHelp.help()'
WASX7031I: For help, enter: "print Help.help()"
wsadmin>
```

3.1.3.3 Using the Oracle Fusion Middleware wsadmin Command-Line Online Help

The following sections describe some key features of the Oracle Fusion Middleware wsadmin command-line shell:

- Listing the Oracle Fusion Middleware wsadmin Command Categories
- Listing the Commands Within an Oracle Fusion Middleware wsadmin Command-Line Category
- Getting Help on a Specific Oracle Fusion Middleware wsadmin Command

3.1.3.3.1 Listing the Oracle Fusion Middleware wsadmin Command Categories To list the available categories of Oracle Fusion Middleware commands in the Oracle Fusion Middleware wsadmin command-line shell, use the following command:

```
wsadmin>print OracleHelp.help()
```

Example 3–2 shows an example of the output of the `print OracleHelp.help()` command when you run it from the Oracle Common home.

If you run the command from an Oracle Fusion Middleware component Oracle home (for example, an Oracle SOA Suite, Oracle WebCenter Portal, or Oracle WebCenter Content Oracle home), then the output will include information on the component-specific wsadmin commands.

Example 3–2 Listing the Available Commands from the Oracle Fusion Middleware wsadmin Command-Line Shell

```
wsadmin>print OracleHelp.help()

ADFMAdmin                   ADFM Lifecycle Management Commands.
MDSAdmin                    MDS Lifecycle Management Commands.
OracleDFW                   Lists commands for FMW diagnostic framework.
OracleDMS                   Lists commands for FMW performance metrics and
                            events.
OracleHelp                  Provides help for Oracle modules.
OracleJRF                   Commands for configuring Managed Servers with
                            Oracle Java Required Files (JRF)
OracleLibOVDConfig          List commands for managing OVD configuration
OracleMWConfig              Oracle Middleware Configuration Tool.
OracleMWConfigUtilities     Oracle Middleware Configuration Tool Utilities.
OracleODL                   Lists commands for FMW diagnostic logging.
URLConnection               List Commands for managing ADF Based URL
                            Connections
WebServices                 Lists commands for Oracle WebServices Management.
audit                       Lists commands for Common Audit Framework
igfconfig                   List commands for manageing IGF configuration
opss                        Oracle platform security services Commands.
wsmManage                   Lists commands for Oracle WSM Policy Management.
```

```
wsadmin>
```

3.1.3.3.2 Listing the Commands Within an Oracle Fusion Middleware wsadmin Command-Line Category

To list the commands associated with a particular category, enter the category name inside single quotation marks within the parentheses. For example:

```
wsadmin>print OracleHelp.help('OracleODL.help')
```

Example 3–3 shows an example of listing the commands in a particular category.

Example 3–3 Listing a Specific Category of Oracle Fusion Middleware wsadmin Commands

```
wsadmin>print OracleHelp.help('OracleODL')

Commands for FMW diagnostic logging

configureLogHandler     Configure Java logging handlers.
displayLogs             Search and display the contents of diagnostic log
                        files.
getLogLevel             Returns the level of a given Java logger.
listLogHandlers         Lists Java log handlers configuration.
listLoggers             Lists Java loggers and their levels.
listLogs                Lists log files for FMW components.
setLogLevel             Sets the level of a given Java logger.

wsadmin>
```

3.1.3.3.3 Getting Help on a Specific Oracle Fusion Middleware wsadmin Command

To get help on a specific Oracle Fusion Middleware wsadmin command:

```
wsadmin>print OracleHelp.help(category.command)
```

Example 3–4 shows an example of the online help output for a specific Oracle Diagnostic Logging command.

Example 3–4 Example of Online Help for a Specific Oracle Fusion Middleware wsadmin Command

```
wsadmin>print OracleHelp.help('OracleODL.listLogs')

Lists log files for FMW components.

Returns a PyArray with one element for each log. The elements of the
array are javax.management.openmbean.CompositeData objects describing
each log.

Syntax:
listLogs([options])
- options: optional list of name-value pairs.

  o target: the name of a Weblogic server, or an OPMN managed FMW component.
    For an OPMN managed component the syntax for the target is
    "opmn:<instance-name>/<component-name>".

    The target argument can be an array of strings containing one or more
    targets. In connected mode the default target includes all running
    Weblogic servers in the domain that have JRF enabled.
    In disconnected mode there is no default, the target option is required.
```

```
  o oracleInstance: defines the path to the ORACLE_INSTANCE (or Weblogic
    domain home). The command will be executed in disconnected mode when
    this parameter is used.

  o unit: defines the unit to use for reporting file size. Valid
    values are B (bytes), K (kilobytes), M (megabytes), G (gigabytes),
    or H (display size in a human-readable form, similar to Unix's "ls
    -h" option). The default value is H.

  o fullTime: a Jython Boolean value. If true, reports the full time
    for the log file last modified time. Otherwise displays a short
     version of the time. The default value is false.

Example:
1. listLogs()
2. listLogs(target="server1")
3. listLogs(target="opmn:instance1/ohs1")
4. listLogs(oracleInstance="/middleware/user_projects/domains/base_domain",
target="server1")
wsadmin>
```

3.1.3.4 Differences Between the wsadmin Commands and the WebLogic Scripting Tool (WLST) Commands

Many of the Oracle Fusion Middleware wsadmin commands that are supported for IBM WebSphere have equivalent WebLogic Scripting Tool (WLST) commands.

To find information about the equivalent WLST command, refer to the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

To list all the Oracle Fusion Middleware wsadmin command categories (or modules) use the `OracleHelp.help()` command, as shown in Example 3–2.

In many cases, the only difference between the WLST command and the wsadmin command is that you must prefix each wsadmin command with the category name. Example 3–6 shows how you might use the `listLoggers` command in WLST. Example 3–7 shows the same command in wsadmin.

Example 3–5 Using the ListLoggers Command in WLST

```
wls:/base_domain/serverConfig> listLoggers(pattern="oracle.dms.*")
-----------------------+-----------------
Logger                 | Level
-----------------------+-----------------
oracle.dms             | <Inherited>
oracle.dms.aggregator  | <Inherited>
oracle.dms.collector   | <Inherited>
oracle.dms.context     | <Inherited>
oracle.dms.event       | <Inherited>
oracle.dms.instrument  | <Inherited>
oracle.dms.jrockit.jfr | <Inherited>
oracle.dms.reporter    | <Inherited>
oracle.dms.trace       | <Inherited>
oracle.dms.translation | <Inherited>
oracle.dms.util        | <Inherited>
wls:/base_domain/serverConfig>
```

Example 3–6 Using the ListLoggers Command in Wsadmin

```
wsadmin>OracleODL.listLoggers(pattern="oracle.dms.*")
-----------------------+-----------------
```

```
Logger                 | Level
-----------------------+-----------------
oracle.dms             | WARNING:1
oracle.dms.aggregator  | NOTIFICATION:1
oracle.dms.collector   | NOTIFICATION:1
oracle.dms.context     | NOTIFICATION:1
oracle.dms.event       | NOTIFICATION:1
oracle.dms.instrument  | NOTIFICATION:1
oracle.dms.reporter    | NOTIFICATION:1
oracle.dms.trace       | NOTIFICATION:1
oracle.dms.translation | NOTIFICATION:1
oracle.dms.util        | NOTIFICATION:1
wsadmin>
```

3.1.3.5 Differences Between Oracle Fusion Middleware wsadmin Commands and IBM WebSphere Wsadmin Commands

Note the following difference between running Oracle Fusion Middleware wsadmin commands and the standard IBM WebSphere wsadmin commands:

- You must run the Oracle Fusion Middleware commands from the `common/bin` directory of the Oracle Fusion Middleware Oracle home.
- The Oracle Fusion Middleware wsadmin commands use the Jython scripting language exclusively.

3.2 Basic Administration Tasks on IBM WebSphere

The following sections provide information about some basic administration tasks you can perform when running Oracle Fusion Middleware on IBM WebSphere:

- Referring to IBM WebSphere DIrectory Paths on Windows Systems
- Starting and Stopping Servers on IBM WebSphere
- Configuring Metadata Services (MDS) on IBM WebSphere
- Configuring Oracle Fusion Middleware Logging on IBM WebSphere
- Setting Up the Diagnostic Framework
- Creating a Data Source in an IBM WebSphere Cell

3.2.1 Referring to IBM WebSphere DIrectory Paths on Windows Systems

If you are providing the path to the WebSphere Application Server on a Windows operating system, and a directory name in the path includes a space, you need to supply a shortened name, with a tilde character (~) followed by a `1` instead of the character before the space.

For example, the default location of a WebSphere Application Server on a Windows operation system is in a subdirectory of Program Files, a directory name that includes a space:

```
C:\Program Files\IBM\WebSphere\Appserver
```

This location needs to be specified as follows:

```
C:\Progra~1\IBM\WebSphere\Appserver
```

If you are browsing to this location, the **Browse** button incorrectly populates the field with the space rather than `C:\Progra~1`.

3.2.2 Starting and Stopping Servers on IBM WebSphere

There are two methods for starting and stopping the servers in your IBM WebSphere cell:

- Starting and Stopping IBM WebSphere Servers with Profile Scripts
- Starting and Stopping IBM WebSphere Servers with Fusion Middleware Control

3.2.2.1 Starting and Stopping IBM WebSphere Servers with Profile Scripts

Just as with any other IBM WebSphere cell, you can use profile scripts to start and stop the servers in a cell you configured for Oracle Fusion Middleware.

For example, to stop the OracleAdminServer, navigate to the following directory in the IBM WebSphere home, and enter the following command:

On UNIX operating systems:

```
profiles/profile_name/bin/stopServer.sh OracleAdminServer
         -profileName profileName
```

On Windows operating systems:

```
profiles\profile_name\bin\stopServer.cmd OracleAdminServer
         -profileName profileName
```

For example, on a UNIX operating system:

```
/disk01/IBM/WebSphere/ApplicationServer/profiles
        /Custom01/bin/stopServer.sh OracleAdminSErver
         -profileName Custom01
```

For examples of how to start the servers in your IBM WebSphere cell, see Section 2.8, "Task 8: Start the IBM WebSphere Servers".

For more information about the scripts that are generated for each profile, refer to the IBM WebSphere documentation.

3.2.2.2 Starting and Stopping IBM WebSphere Servers with Fusion Middleware Control

You can also stop and start IBM WebSphere servers from Oracle Enterprise Manager Fusion Middleware Control.

For example, to stop a server from Fusion Middleware Control:

1. Navigate to the Server home page.

   For more information, see Section 3.1.2.5, "Viewing an IBM WebSphere Server from Fusion Middleware Control".

2. From the **WebSphere Application Server** menu, select **Control**, and then select **Shut down**.

   Fusion Middleware Control displays a confirmation dialog box.

3. Click **Shutdown**.

Note: Fusion Middleware Control is deployed to the OracleAdminServer. As a result, if you stop the OracleAdminServer, then Fusion Middleware Control will be stopped, and you must use the profile scripts to start the servers.

For more information, see Section 3.2.2.1, "Starting and Stopping IBM WebSphere Servers with Profile Scripts".

3.2.3 Configuring Metadata Services (MDS) on IBM WebSphere

On IBM WebSphere, you can manage Oracle Fusion Middleware Metadata Services (MDS) using Oracle Enterprise Manager Fusion Middleware Control and the `wsadmin` command-line utility, just as you can other Oracle Fusion Middleware components.

Refer to the following sections for more information about the differences from configuring MDS on Oracle WebLogic Server:

- Differences in MDS Command-Line Features on IBM WebSphere
- Differences in MDS Fusion Middleware Control Pages on IBM WebSphere

3.2.3.1 Differences in MDS Command-Line Features on IBM WebSphere

All the wsadmin commands you use to manage MDS on IBM WebSphere have equivalent WebLogic Scripting Tool (WLST) commands, which are documented in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

In addition, refer to the wsadmin online help for information about any differences between the MDS commands available in WLST and in wsadmin.

For example, note the following differences when using the `registerMetadataDBRepository` command on IBM WebSphere:

- The command has an additional parameter on IBM WebSphere (`authAlias`).
- The existing `targetServers` parameter allows you to specify a target WebSphere server or cluster for the repository, rather than a Oracle WebLogic Server instance.

For more information, see the following:

- Using the registerMetadataDBRepository authAlias parameter on IBM WebSphere.
- Using the registerMetadataDBRepository targetServers Parameter on IBM WebSphere
- More Information About the registerMetadaDBRepository Command on IBM WebSphere

3.2.3.1.1 Using the registerMetadataDBRepository authAlias parameter on IBM WebSphere

Use the `authAlias` argument to create or use an existing authentication alias for connecting to the database where the MDS schema resides. For example:

- If you do not provide a value for the `authAlias` parameter, then Oracle Fusion Middleware assumes that the authentication alias name is the same as the metadata repository name.
- If you provide a user name and password, then Oracle Fusion Middleware creates a new authentication alias either by using the value of the `authAlias` parameter as the alias name if it is provided, or by using the name of the metadata repository as alias name if the `authAlias` parameter is not provided.

- If you do not provide a user name and password, then Oracle Fusion Middleware assumes you want to connect to the database using the existing authentication alias, which is either the value of the `authAlias` parameter or the name of the metadata repository if the `authAlias` parameter is not provided.

3.2.3.1.2 Using the registerMetadataDBRepository targetServers Parameter on IBM WebSphere

Use the `targetServers` parameter to specify the WebSphere servers or clusters to which this repository will be registered. If this argument is not specified, then the repository will be registered only to the DeploymentManager.

The server or cluster must be specified in the form of specifying a configuration object in the wsadmin scripting tool. A configuration object can be specified as multiple `/type:name/` value pairs in the containment path string. For example:

```
'/Cell:myCell/Node:myNode/Server:myServer/'
```

The containment path must be a path that contains the correct hierarchical order.

To specify multiple servers or clusters, separate the names with a comma.

Note that if you later add additional servers or clusters to the cell, you must do one of the following to ensure that the repository is available from the new servers or clusters that were added after the initial registration of the repository:

- Use the `deregisterMetadataDBRepository` command to deregister the repository from all the initial targets, then run the `registerMetadataDBRepository` command again to reregister the repository with more targets. Note that the data source will be unavailable on all servers until you run the `registerMetadataDBRepository` command the second time.
- Manually create the data source on the new servers or clusters, using the exact same properties as the data source you created with the `registerMetadataDBRepository` command.

3.2.3.1.3 More Information About the registerMetadaDBRepository Command on IBM WebSphere

For more information about using the `registerMetadataDBRepository` command on IBM WebSphere, review the wsadmin online help for the command:

```
wsadmin> print MDSAdmin.help('registerMetadataDBRepository')
```

For more information about using wsadmin command-line online help, see Section 3.1.3.3, "Using the Oracle Fusion Middleware wsadmin Command-Line Online Help".

3.2.3.2 Differences in MDS Fusion Middleware Control Pages on IBM WebSphere

When you are using Fusion Middleware Control to manage the MDS repository on IBM WebSphere, there are some differences in the Fusion Middleware Control pages. These differences are due to the differences in the basic administration functions for Oracle WebLogic Server and IBM WebSphere.

For example:

- On Oracle WebLogic Server, the Metadata Repository home page includes a Targeted Servers region, which identifies Oracle WebLogic Server servers that can access the repository. This region is not available on IBM WebSphere.
- On IBM WebSphere, the Register Database-Based Metadata Repository page provides the ability to specify an authentication alias, which can be used to represent the credentials required to connect to the repository database.

3.2.4 Configuring Oracle Fusion Middleware Logging on IBM WebSphere

There are several ways to change the configuration of log files for the Oracle Fusion Middleware products when running with IBM WebSphere.

Consider the following when modifying the log configuration:

- To change the log levels, you can use the IBM WebSphere Administrative Console, Fusion Middleware Control, or the `OracleODL` commands in the Oracle Fusion Middleware wsadmin command-line shell.

  Note that in IBM WebSphere, `java.util.logging` is implemented differently than in Oracle WebLogic Server; specifically, child loggers do not inherit the log level property from the parent. However, you can change the log levels for a logger and its descendants, by using the wsadmin commands shown in Example 3–7.

  Note that in Example 3–7, the two spaces before the `OracleODL.setLogLevel` command are required. The spaces indicate that this line is a continuation of the previous line.

- To change other configuration properties, you can use Fusion Middleware Control, or the `OracleODL` commands in the wsadmin command line.

- The name of the log configuration file is `websphere-logging.xml`. Note, however, that you should not edit the file directly; you should use Fusion Middleware Control, the wsadmin command line, or the IBM WebSphere Administrative Console to modify the file.

- The main diagnostic log file is located in the following directory:

  SERVER_LOG_ROOT`/`*server_name*`-diagnostic.log`

  For more information about the SERVER_LOG_ROOT environment variable, see the IBM WebSphere documentation.

  Note that some Oracle Fusion Middleware components also generate their own logs, which are also stored in this location.

Example 3–7 Sample Oracle Fusion Middleware Wsadmin Script that Sets Logging Levels

```
wsadmin>myLoggers = OracleODL.listLoggers(pattern="oracle.dms.*")
----------------------+-----------------
Logger                | Level
----------------------+-----------------
oracle.dms            | WARNING:1
oracle.dms.aggregator | NOTIFICATION:1
oracle.dms.collector  | NOTIFICATION:1
oracle.dms.context    | NOTIFICATION:1
oracle.dms.event      | NOTIFICATION:1
oracle.dms.instrument | NOTIFICATION:1
oracle.dms.reporter   | NOTIFICATION:1
oracle.dms.trace      | NOTIFICATION:1
oracle.dms.translation | NOTIFICATION:1
oracle.dms.util       | NOTIFICATION:1
wsadmin>print myLoggers
{'oracle.dms.translation': 'NOTIFICATION:1', 'oracle.dms.context':
 'NOTIFICATION:1', 'oracle.dms.event': 'NOTIFICATION:1', 'oracle.dms':
 'NOTIFICATION:1', 'oracle.dms.util': 'NOTIFICATION:1', 'oracle.dms.aggregator':
 'NOTIFICATION:1', 'oracle.dms.reporter': 'NOTIFICATION:1', 'oracle.dms.trace':
 'NOTIFICATION:1', 'oracle.dms.instrument': 'NOTIFICATION:1',
 'oracle.dms.collector': 'NOTIFICATION:1'}
```

```
wsadmin> for loggerName in myLoggers.keys():
wsadmin>  OracleODL.setLogLevel(target="OracleAdminServer", logger=loggerName,
level="FINE")
wsadmin>
wsadmin>OracleODL.listLoggers(pattern="oracle.dms.*")
----------------------+-----------------
Logger                | Level
----------------------+-----------------
oracle.dms            | WARNING:1
oracle.dms.aggregator | TRACE:1
oracle.dms.collector  | TRACE:1
oracle.dms.context    | TRACE:1
oracle.dms.event      | TRACE:1
oracle.dms.instrument | TRACE:1
oracle.dms.reporter   | TRACE:1
oracle.dms.trace      | TRACE:1
oracle.dms.translation | TRACE:1
oracle.dms.util       | TRACE:1
```

3.2.5 Setting Up the Diagnostic Framework

Because the Automatic Diagnostic Repository (ADR) binaries are not automatically installed when Oracle Fusion Middleware is installed on IBM WebSphere, the Diagnostic Framework cannot access the ADR to store incidents.

To allow incident creation on IBM WebSphere, you must install the ADR binaries and configure each WebSphere server to point to those binaries.

Perform the following steps:

1. Download and install the Oracle Database Instant Client binaries version 11.2.0.1 from Oracle Technology Network (OTN).

   http://www.oracle.com/technology/software/tech/oci/instantclient/index.html

   Select your operating system, then select **Basic.**

2. Install the downloaded files on the host on which IBM WebSphere is running.

3. Configure the IBM Websphere server to set the system property `oracle.adr.home` to the location of the installed Oracle Database Instant Client binaries, using the WebSphere Integrated Solutions Console.

   For example, to set the property on distributed platforms:

   a. Expand **Servers,** then **Server Types.** Select **WebSphere application servers.**

   b. On the Application servers page, select the server.

   c. In the Server Infrastructure section of the server page, expand **Java and process management,** then select **Process Definition.**

   d. In the Process Definition page, select **Java Virtual Machine.**

   e. Select **Custom Properties,** then click **New.**

   f. For **Name,** enter `oracle.adr.home`.

   g. For **Value,** enter the location of the installed files.

   h. Click **Apply,** then **Save.**

3.2.6 Creating a Data Source in an IBM WebSphere Cell

Creating a data source is a common administration task, which is required when configuring certain aspects of your Oracle Fusion Middleware environment.

Data sources that connect to the product schemas installed by the Repository Creation Utility are created when you run the Configuration Wizard. However, there are other scenarios where you might need to create a data source--for example, you might need a data source for the applications you deploy.

To create a data source on IBM WebSphere, you can use the IBM WebSphere Administrative Console.

The following example shows how to create an IBM WebSphere data source for an Oracle database. Creating the database involves the following tasks:

- Task 1, "Create an authentication alias for the Oracle database you want to access"
- Task 2, "Create a JDBC data provider for the Oracle database"
- Task 3, "Modify the JDBC data provider to use the latest Oracle database classes"
- Task 4, "Create a JDBC data source that uses the Oracle database JDBC provider"
- Task 5, "Test the Data Source Connection"

Task 1 Create an authentication alias for the Oracle database you want to access

1. Log in to the IBM WebSphere Administrative Console and navigate to **Security** > **Global Security**.
2. On the Global Security page, select **Java Authentication and Authorization Service** > **J2C Authentication Data**.
3. Click **New**.
4. On the General Properties page enter the information shown in Table 3–2.
5. Save the new authentication alias to the master configuration.

Table 3–2 Authentication Alias General Properties for an Oracle Database Data Source

Element	Description
Alias	Enter a name for the alias. Use a name that identifies the purpose of the credentials assigned to the alias. For example, `OracleDBalias`.
User ID	Enter the Oracle database user name you will use to connect to the database. **Note:** Where required, also include the role. For example, if you are connecting as SYS, then enter the following in this field: `SYS as SYSDBA`
Password	Enter the password for the database user.
Description	Optionally, enter a description that describes the purpose of the authentication alias.

Task 2 Create a JDBC data provider for the Oracle database

1. Log in to the IBM WebSphere Administrative Console and navigate to **Resources > JDBC > JDBC Providers**.
2. Select the appropriate **Scope** for the data provider you are about to create.

3. Click **New**.

   The IBM WebSphere Administrative Console displays a three-step wizard to guide you through the JDBC provider creation process.

4. In Step 1 of the JDBC provider wizard, make the selections shown in Table 3–3.

5. In Step 2 of the JDBC provider wizard, accept the default values.

   Note: You will modify these later in the procedure.

6. In Step 3 of the JDBC provider wizard, verify the values you entered and selected so far.

7. Click **Finish** to create the initial provider and return to the JDBC Providers page.

Table 3–3 Recommended Values to Select When Creating an IBM WebSphere Data Source for an Oracle Database

Element	Recommended Value
Database Type	Select **Oracle** from the drop-down menu.
Provider Type	Select **Oracle JDBC Driver** from the drop-down menu.
Implementation Type	Select **Connection pool data source** from the drop-down menu.
Name	Provide a unique name for the JDBC provider, or use the default name.
Description	Optionally, provide a description of the JDBC provider. This can be useful if you are creating multiple data sources for specific purposes.

Task 3 Modify the JDBC data provider to use the latest Oracle database classes

1. Click the name of the database provider in the list of JDBC providers.

2. In the General properties section of the page, replace the value in the **Class path** field with the following:

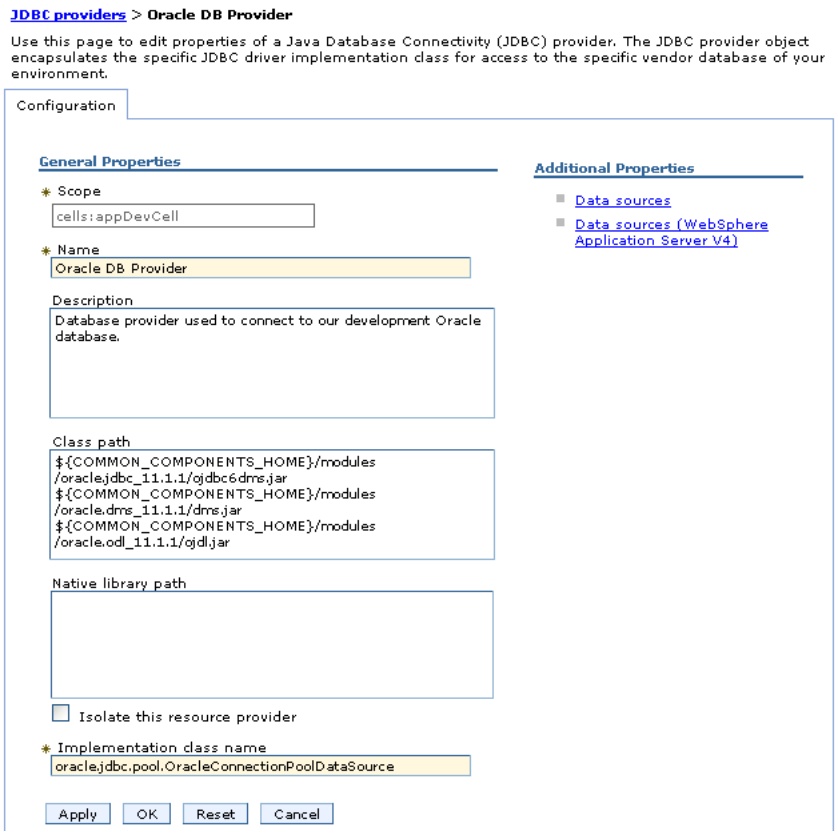
```
${COMMON_COMPONENTS_HOME}/modules/oracle.jdbc_11.1.1/ojdbc6dms.jar
${COMMON_COMPONENTS_HOME}/modules/oracle.dms_11.1.1/dms.jar
${COMMON_COMPONENTS_HOME}/modules/oracle.odl_11.1.1/ojdl.jar
```

   Press Enter to separate the path locations so they appear on one line each, as shown in Figure 3–2.

3. Click **OK** to return to the JDBC Providers page.

4. Click **Save** to save your changes to the master configuration.

Figure 3–2 Summary of IBM WebSphere JDBC Provider Values for an Oracle Database



Task 4 Create a JDBC data source that uses the Oracle database JDBC provider

1. Log in to the console and navigate to **Resources > JDBC > Data Sources**.
2. Select the appropriate **Scope** for the data source you are about to create.
3. Click **New**.

   The IBM WebSphere Administrative Console displays a five-step wizard to guide you through the data source creation process.
4. In Step 1 of the data source wizard, enter a name for the data source and a JNDI location.

   For example, use `myOracleDS` as the data source name and `jdbc/myOracleDS` as the JNDI location.
5. In Step 2 of the data source wizard, select **Select an existing JDBC provider** and select the JDBC provider you created earlier in this procedure from the drop-down menu.
6. In Step 3 of the data source wizard, do the following:
   a. In the **URL** field, enter the connection string for the Oracle database, using the following format:

      `jdbc:oracle:thin:@`*`hostname`*`:`*`port`*`:`*`SID`*

      For example:

      `jdbc:oracle:thin:@host42.example.com:1521:DB43`

b. From the **Data store helper class name** menu, select the appropriate class name, based on whether you are connecting to a 10*g* or 11*g* Oracle database.

c. Optionally, select **Use this data source in container managed persistence (CMP)**.

   See the IBM WebSphere Administrative Console online help for information about the purpose of this option.

7. In Step 4 of the data source wizard, use the **Component-managed authentication alias** menu to select the authentication alias you created for the Oracle database earlier in this procedure.

   See the IBM WebSphere Administrative Console online help for information about the other options on the page.

8. In Step 5 of the wizard, review your changes. If they are accurate, click **Finish** to return to the Data Sources page.

9. Save the configuration changes, as directed in the console.

Task 5 Test the Data Source Connection

On the Data Sources page, select the data source and click **Test Connection** to verify your data source configuration.

3.3 Deploying Applications on IBM WebSphere

Refer to the following sections for information on deploying your Oracle Fusion Middleware applications on IBM WebSphere:

- Preparing to Deploy Oracle Fusion Middleware Applications on IBM WebSphere
- Methods for Deploying Oracle Fusion Middleware Applications on IBM WebSphere
- Deploying Applications that Require MDS Deployment Plan Customizations on IBM WebSphere

3.3.1 Preparing to Deploy Oracle Fusion Middleware Applications on IBM WebSphere

Before you can deploy Oracle Fusion Middleware applications (such as ADF, Oracle SOA Suite, Oracle WebCenter Portal, or Oracle WebCenter Content applications) to IBM WebSphere, you must follow certain steps for preparing the environment.

For example, you must be sure the Java Required Files (JRF) template has been applied to the IBM WebSphere servers. This can be accomplished by configuring the environment using the Oracle Fusion Middleware Configuration Wizard, as described in Chapter 2, "Installing and Configuring Oracle Fusion Middleware on IBM WebSphere" and in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

3.3.2 Methods for Deploying Oracle Fusion Middleware Applications on IBM WebSphere

The primary methods for deploying your Oracle Fusion Middleware applications to IBM WebSphere are as follows:

- If you are working in a development or testing environment, you can deploy your applications directly from Oracle JDeveloper.

For information about configuring Oracle JDeveloper with an IBM WebSphere environment, see "Deploying the Application" in the *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

For information about deploying Oracle SOA Suite, Oracle WebCenter Portal, or Oracle WebCenter Content applications, refer to the corresponding chapter in this guide and the appropriate product development guide.

- If you are working in a testing or production environment, you can deploy application archives--for example, Enterprise Archive (EAR) files--from the IBM WebSphere Administration Console.

3.3.3 Deploying Applications that Require MDS Deployment Plan Customizations on IBM WebSphere

To deploy an application that requires MDS Deployment Plan customizations, you must use Oracle JDeveloper, unless you use the MDS wsadmin commands to customize the MDS deployment plan.

After you customize the deployment plan, you can then deploy the application archive from the IBM WebSphere Administrative Console.

3.4 Configuring Oracle Fusion Middleware High Availability on IBM WebSphere

The following sections provide information on configuring Oracle Fusion Middleware components for high availability on IBM WebSphere:

- Documentation Resources for Configuring Oracle Fusion Middleware High Availability on IBM WebSphere
- Configuring Java Object Cache for Oracle Fusion Middleware on IBM WebSphere

3.4.1 Documentation Resources for Configuring Oracle Fusion Middleware High Availability on IBM WebSphere

When configuring a high availability environment for the Oracle Fusion Middleware components that you install and configure on IBM WebSphere, refer to the following resources:

- The IBM WebSphere documentation available on the WebSphere Application Server Information Center.
- The *Oracle Fusion Middleware High Availability Guide*, which describes basic high availability concepts for Oracle Fusion Middleware components on Oracle WebLogic Server.
- The Oracle Fusion Middleware Enterprise Deployment Guides, which provide specific reference topologies for configuring the various Oracle Fusion Middleware components in a Oracle WebLogic Server-based production environment.
- Section 5.2.19, "Configuring WebCenter Portal Applications for High Availability on IBM WebSphere" for specific information about configuring Oracle WebCenter Portal for high availability
- The *Oracle Fusion Middleware Release Notes* for your platform, for information about known issues and workarounds when configuring Oracle Fusion Middleware components on IBM WebSphere.

In addition, refer to "Using wsadmin to Configure Oracle Fusion Middleware" in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*, which provides examples of how you can use the wsadmin command-line to:

- Create servers, clusters, and cluster members on IBM WebSphere
- Create data sources for communicating with an Oracle Real Application Clusters database
- Federate remote nodes to an existing cell

3.4.2 Configuring Java Object Cache for Oracle Fusion Middleware on IBM WebSphere

When configuring high availability for Oracle Fusion Middleware, the *Oracle Fusion Middleware High Availability Guide* and Oracle Fusion Middleware Enterprise Deployment Guides suggest using Java Object Cache (JOC) to increase the performance of Oracle Web Services Manager and Oracle WebCenter Portal.

To configure JOC in this scenarios, Oracle Fusion Middleware provides a custom script called `configure-joc.py`. This script is not supported on IBM WebSphere.

As an alternative, you can use the following procedure to configure JOC for Oracle Fusion Middleware on IBM WebSphere:

1. Locate and edit the `javacache.xml` file for each the server in the cluster.

   The javacache.xml file is located in the Deployment Manager directory for each server:

   ```
   WAS_HOME/profiles/dmgr_proile_name/config
           /cells/cell_name
           /nodes/node_name
           /servers/server_name
           /fmwconfig/javacache.xml
   ```

   For example, if you have configured a cluster called WC_Spaces, and the cluster contains two servers, WC_Spaces and WC_Spaces2, then you can locate the `javacache.xml` file as follows:

   ```
   WebSphere/AppServer/profiles/Dmgr01/config/cells/Cell01/nodes/Node01/servers/WC
   _Spaces/fmwconfig/javacache.xml

   WebSphere/AppServer/profiles/Dmgr01/config/cells/Cell01/nodes/Node01/servers/WC
   _Spaces2/fmwconfig/javacache.xml
   ```

2. Make the following changes to the `javacache.xml` file:

   - Set the `enabled` attribute of the `<communicationService>` element to TRUE.
   - Remove the `outOfProc="false"` attribute from the `<packet-distributer>` element.
   - Add the `<distributor-location>` elements with the host and port of the servers in the cluster

   Example 3–8 provides a sample `javacache.xml` file that has been modified for use on IBM WebSphere. In the example, replace host with the host address and replace port with the port used for JOC communication. You can select any free port.

3. Login to the IBM WebSphere Administrative Console and navigate to the Nodes page (**System administration > Nodes**).

4. Select all nodes in the cluster and click on **Full Resynchronize**.
5. Restart all servers in the cluster.

Example 3–8 Sample javacache.xml File - Modified for IBM WebSphere

```
<?xml version="1.0" encoding="UTF-8"?>
<cache-configuration
    xmlns="http://www.oracle.com/oracle/ias/cache/configuration11"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" max-objects="5000"
    max-size="10" private="false"
    cache-dump-path="jocdump" system="false" clean-interval="60"
    version="11.1.1.2.0"
    internal-version="110000">
  <communicationService enabled="true">
    <v2 ssl-config-file=".sslConfig" init-retry="300" init-retry-delay="2000"
        enable-ssl="false" auto-recover="false">
      <packet-distributor enable-router="false" startable="true"
        dedicated-coordinator="false" >
        <distributor-location host="myhost1.example.com"
          port="9988" ssl="true"/>
       <distributor-location host="myhost2.exmaple.com" port="9988"
          ssl="true"/>
      </packet-distributor>
    </v2>
  </communicationService>
  <diskCache size="10" count="5000" ping-interval="60"/>
  <logging override-parent="false" location="javacache.log"
     default-level="SEVERE"/>
  <dms enabled="false"/>
</cache-configuration>
```

4

Managing Oracle SOA Suite on IBM WebSphere

This chapter contains information about managing Oracle SOA Suite applications and components on IBM WebSphere.

This chapter contains the following sections:

- Configuring Oracle SOA Suite and Oracle BAM Against an External LDAP Server on IBM WebSphere
- Differences and Restrictions When Developing and Deploying Oracle SOA Suite Applications on IBM WebSphere
- Differences and Restrictions When Managing Oracle SOA Suite Components on IBM WebSphere
- Differences and Restrictions When Managing Oracle BAM on IBM WebSphere

In this chapter, IBM WebSphere is used to reference both IBM WebSphere Application Server (AS) and IBM WebSphere Application Server Network Deployment (ND). The specific product names are used when appropriate.

4.1 Configuring Oracle SOA Suite and Oracle BAM Against an External LDAP Server on IBM WebSphere

If you are installing and configuring Oracle SOA Suite on IBM WebSphere, then you must install and configure a supported LDAP server before you can configure the Oracle SOA Suite components in a new IBM WebSphere cell. For more information, see Section 4.1.1, "Configuring SOA Suite Users and Groups in an External LDAP Server."

If you are installing Oracle BAM on IBM WebSphere, then you must perform additional configuration steps for Oracle SOA Suite and Oracle BAM against the external LDAP server. For more information, see Section 4.1.2, "Configuring Oracle SOA Suite and Oracle BAM in an External LDAP Server."

4.1.1 Configuring SOA Suite Users and Groups in an External LDAP Server

When you install Oracle SOA Suite with IBM WebSphere, an internal LDAP server is *not* automatically configured with SOA users and groups. You must manually perform these configuration tasks in an external LDAP server, such as Oracle Internet Directory, after installation.

For information on the LDAP servers that are supported by Oracle Fusion Middleware, refer to the certification information on the Oracle Technology Network:

```
http://www.oracle.com/technology/software/products/ias/files/fusion_
certification.html
```

The following provides an overview of the tasks to perform when configuring your supported LDAP server for use with Oracle SOA Suite:

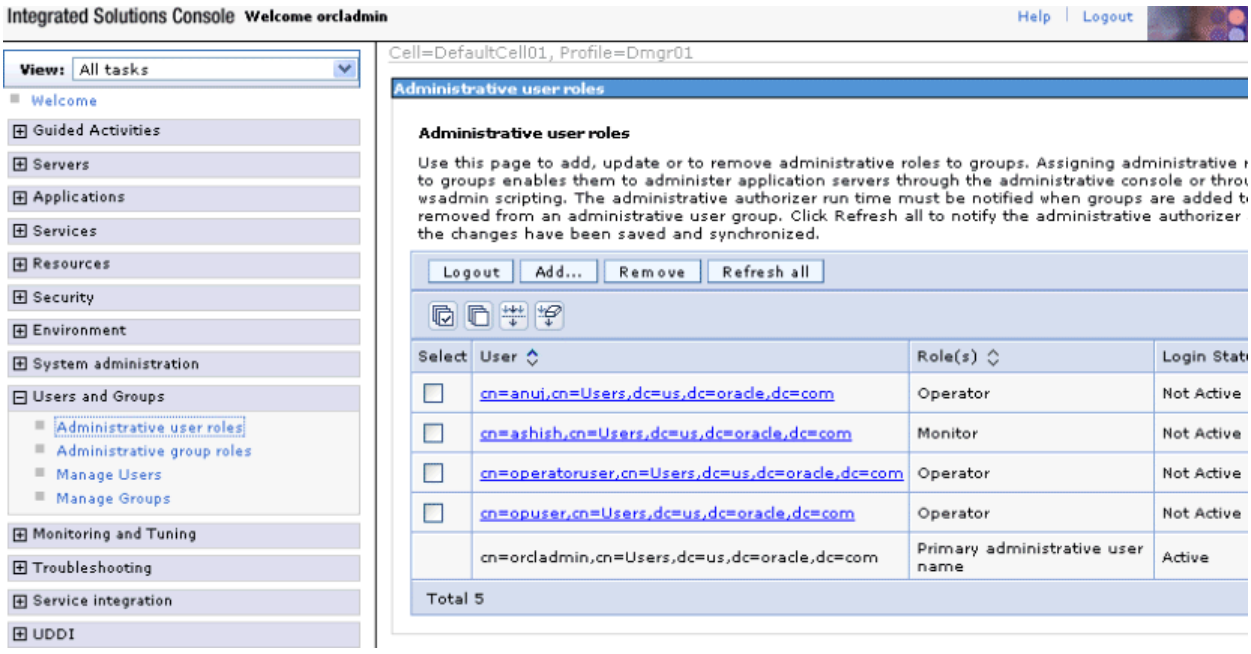
1. Use your LDAP management tool to create two groups (Operator group and Monitor group) and two users (Operator user and Monitor user).

   Note that the management tool you use to create the users and groups will vary, depending up on the LDAP server you are using. For example, if you are using Oracle Internet Directory, refer to information about using the Oracle Directory Services Manager in the *Oracle Fusion Middleware Administrator's Guide for Oracle Internet Directory*.

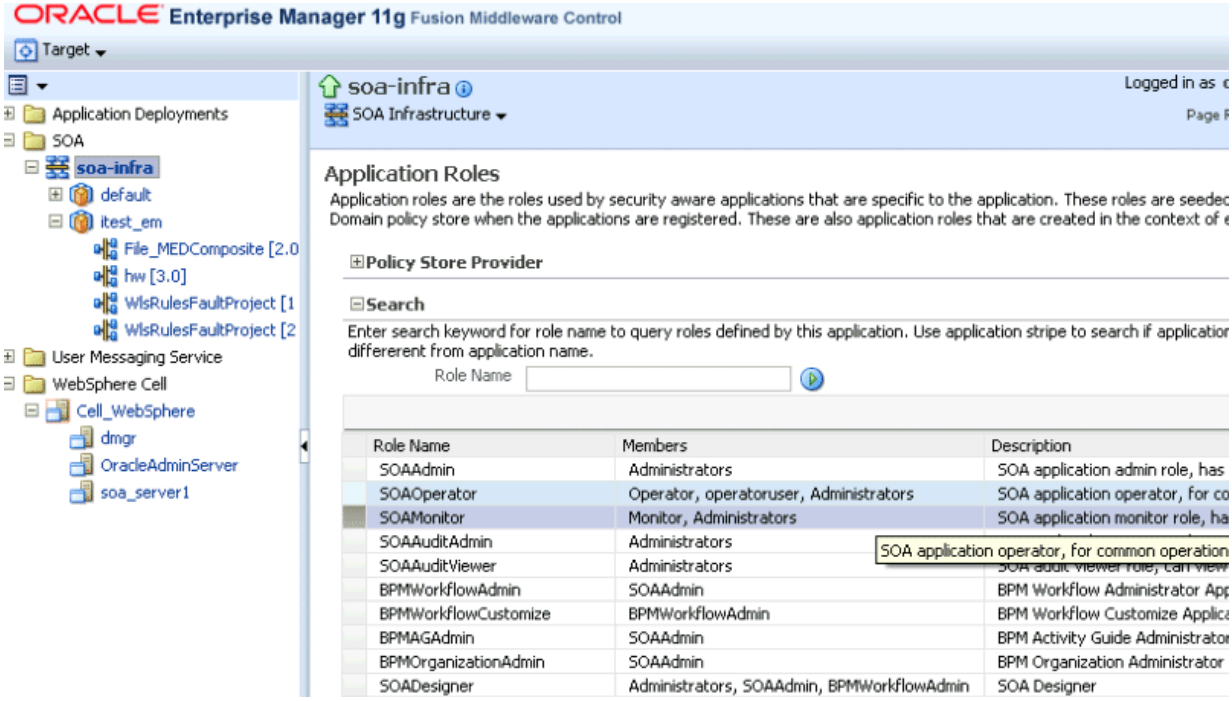
2. In the IBM WebSphere Administrative Console, create the following mappings:

   - User roles for operator and monitor
   - Group roles for operator and monitor

   For example, the following page shows the **Administrative user roles** section with the monitor user **ashish** (second check box) and the operator user **opuser** (fourth check box) available for selection. You perform similar mappings for group roles on a separate page.



3. Log in to Oracle Enterprise Manager Fusion Middleware Control with administrator access.
4. In the navigator, right-click **soa-infra**, and select **Security** > **Application Roles**.
5. Map the SOA roles to the **Operator** and **Monitor** roles.
   - For **SOAOperator** role, add the **Operator** group as a member.
   - For **SOAMonitor** role, add the **Monitor** group as a member.



For additional information about switching LDAP authentication providers, see the following documentation:

- To switch LDAP authentication providers if the corresponding LDAP server contains the user or users who start the domain, see Section "Requirements for Using an LDAP Authentication Provider" in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.
- To add an Oracle Internet Directory, Oracle Virtual Directory, or other authentication provider using WLST commands, see Section "Configuring Additional Authentication Providers" in *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*.

4.1.2 Configuring Oracle SOA Suite and Oracle BAM in an External LDAP Server

To use the external LDAP server with Oracle BAM on IBM WebSphere, the user `OracleSystemUser` must be added to the external LDAP server.

In addition, the following post-installation steps must be executed on IBM WebSphere:

1. Create the properties file to use as the input for configuring the identity store. For example, the Oracle Internet Directory properties file could look like this:

```
user.search.bases=dc=com
group.search.bases=dc=com
subscriber.name=dc=com
ldap.host=mymachine.us.oracle.com
ldap.port=17234
admin.id=cn=orcladmin
admin.pass=orcladmin1
user.filter=(&(cn=%v)(objectclass=person))
group.filter=(&(cn=%v)(objectclass=groupofuniquenames))
user.id.map=*:cn
group.id.map=*:cn
group.member.id.map=groupofuniquenames:uniquemember
ssl=false
```

2. Go to the *MW_HOME*/oracle_common/common/bin directory, where *MW_HOME* is the directory in which Oracle SOA Suite is installed. Then run:

   ```
   ./wsadmin.sh -conntype SOAP -user <username> -password <password>
   ```

   Replace <username> and <password> with the WebSphere user and password for your IBM WebSphere installation.

3. Run the following command to configure the identity store:

   ```
   Opss.configureIdentityStore(propsFileLoc="(<complete_path_LDAP.properties>")
   ```

   For example:

   ```
   Opss.configureIdentityStore(propsFileLoc="C:\oid.properties")
   ```

4. Then run the following command to reassociate the identity store:

   ```
   Opss.reassociateSecurityStore(domain="WAS_policy_
   store",admin="<LDAPAdminUser>",
   password="<LDAPAdminpassword>", ldapurl="ldap://<LDAPHost>:<LDAPPort>",
   servertype="<LDAPSERVERTYPE>", jpsroot="cn=jpsroot")
   ```

   For example:

   ```
   Opss.reassociateSecurityStore(domain="WAS_policy_store",admin="cn=orcladmin",
   password="orcladmin1",
   ldapurl="ldap://mymachine.us.oracle.com:17234", servertype="OID",
   jpsroot="cn=jpsroot")
   ```

4.2 Differences and Restrictions When Developing and Deploying Oracle SOA Suite Applications on IBM WebSphere

The following sections describe differences and restrictions when developing and deploying Oracle SOA Suite applications on IBM WebSphere:

- Section 4.2.1, "Oracle SOA Suite wsadmin and WLST Command Differences"
- Section 4.2.2, "Configuring the WebSphere Application Client for Use with Oracle JDeveloper"
- Section 4.2.3, "Configuring the Proxy on IBM WebSphere Server"
- Section 4.2.4, "Creating an Application Server Connection"
- Section 4.2.5, "Deploying SOA Composite Applications"
- Section 4.2.6, "Using EJB Bindings"
- Section 4.2.7, "AQ Technology Adapter and WebSphere 7.0"
- Section 4.2.8, "JMS Technology Adapter on WebSphere 7.0"
- Section 4.2.9, "Oracle Database Adapter on WebSphere 7.0"

4.2.1 Oracle SOA Suite wsadmin and WLST Command Differences

All Oracle SOA Suite wsadmin commands supported by IBM WebSphere have equivalent WebLogic Scripting Tool (WLST) commands. Table 4–1 describes differences between wsadmin and WLST.

Table 4–1 Differences Between wsadmin and WLST

Issue	WLST	wsadmin
wsadmin command line syntax	WLST commands are prefixed with sca_. For example: `sca_deployComposite('http://adc10:9080','/tmp/sca_HelloWorld_rev1.0.jar')`	All wsadmin commands are prefixed with "soa." to the front of sca_. For example: `soa.sca_deployComposite('http://adc10:9080', '/tmp/sca_HelloWorld_rev1.0.jar')`
Boolean type	You use true/false or 1/0.	You must use 1/0.

Composite management commands	You run WLST commands in offline mode.	You run wsadmin commands in online mode. Command names and signatures are slightly different from WLST commands:
		■ Mb is attached to the end of the command.
		■ Signatures do not include properties for host, port, user, or password.
		To start a composite:
		`soa.sca_startCompositeMb(compositeName, revision, label, partition)`
		To stop a composite:
		`soa.sca_stopCompositeMb(compositeName, revision, label, partition)`
		To activate a composite:
		`soa.sca_activateCompositeMb(compositeName, revision, label, partition)`
		To retire a composite:
		`soa.sca_retireCompositeMb(compositeName, revision, label, partition)`
		To assign a default composite:
		`soa.sca_assignDefaultCompositeMb(compositeName, revision, partition)`
		To get a default composite revision:
		`soa.sca_getDefaultCompositeRevisionMb(compositeName, partition)`
		To list deployed composites:
		`soa.sca_listDeployedCompositesMb()`
		To list all composites in the given partition:
		`soa.sca_listCompositesInPartitionMb(partition)`

Note: wsadmin online commands using MBeans may not provide specific error details. Instead, you may see just an MBeanException.

Execute Oracle SOA Suite wsadmin commands from the *SOA_ORACLE_HOME*/common/bin directory:

```
cd SOA_ORACLE_HOME/common/bin
./wsadmin.sh
```

To invoke online help for Oracle SOA Suite commands, enter the following:

```
wsadmin> print OracleHelp.help('soa')
```

To invoke online help for a specific command, enter the following:

```
wsadmin> print OracleHelp.help('soa.sca_deployComposite')
```

For more information about `wsadmin` commands, see Section 3.1.3, "Using the Oracle Fusion Middleware wsadmin Commands".

For information about the equivalent Oracle SOA Suite WLST commands, see *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

4.2.2 Configuring the WebSphere Application Client for Use with Oracle JDeveloper

This section describes how to configure the WebSphere Application Client for use with Oracle JDeveloper. Once the WebSphere Application Client is properly configured, Oracle JDeveloper can remotely connect to an IBM WebSphere Server. This enables you to perform actions such as the following in Oracle JDeveloper:

- Remote deployment of SOA composite applications and J2EE applications
- Browsing of SOA composite applications on a remote server

4.2.2.1 Installing the WebSphere Application Client

1. Follow the WebSphere Application Client installation steps provided in the IBM documentation.
2. When selecting WebSphere Application Client features for installation, ensure that you select the following components when prompted:
   - IBM Developer Kit
   - Standalone thin clients and resource adapters
3. Apply the latest fix packs through the IBM Update Installer.

   For more information, see Section 1.3.1, "Supported IBM WebSphere Application Servers".

4.2.2.2 Creating the wsadmin.sh/bat File

1. Make a copy of the example file provided in the instructions at the WebSphere Application Server Information Center that describe how to run the `wsadmin` tool remotely in a Java 2 Platform, Standard Edition environment.
2. Edit the `wsadmin.sh` file (for Linux) or the `wsadmin.bat` file (for Windows) as follows:
   a. Set the `WAS_HOME` variable to your WebSphere Application Client home directory:

On...	Set...
Linux	`WAS_HOME=/home/user/IBM/WebSphere/AppClient`
Windows	`set WAS_HOME=C:\IBM\WebSphere\AppClient`

   b. Set the `USER_INSTALL_ROOT` variable to `WAS_HOME`:

On...	Set...
Linux	`USER_INSTALL_ROOT=${WAS_HOME}`
Windows	`set USER_INSTALL_ROOT=%WAS_HOME%`

c. Set the `wsadminHost` variable to your remote IBM WebSphere Application Server host name:

On...	Set...
Linux	`wsadminHost=-Dcom.ibm.ws.scripting.host=www.example.com`
Windows	`set wsadminHost=-Dcom.ibm.ws.scripting.host=www.example.com`

d. Set the `wsadminPort` variable to your remote IBM WebSphere Server SOAP connector port:

On...	Set...
Linux	`wsadminPort=-Dcom.ibm.ws.scripting.port=8879`
Windows	`set wsadminPort=-Dcom.ibm.ws.scripting.port=8879`

e. Edit the `C_PATH` variable to use the WebSphere Application Client JAR files:

On...	Set...

Linux	`C_PATH="${WAS_HOME}/properties:${WAS_HOME}`
	`/runtimes/com.ibm.ws.admin.client_7.0.0.jar:${WAS_HOME}`
	`/plugins/com.ibm.ws.security.crypto.jar"`

Windows	`set C_PATH="%WAS_HOME%\properties;%WAS_`
	`HOME%\runtimes\com.ibm.ws.admin.client_7.0.0.jar;%WAS_`
	`HOME%\plugins\com.ibm.ws.security.crypto.jar"`

f. If installing on Windows, perform the following modifications to the `wsadmin.bat` file.

a. Add `@setlocal` to the beginning of the file.

b. Replace the following code:

```
if exist "%JAVA_HOME%\bin\java.exe" (
    set JAVA_EXE="%JAVA_HOME%\bin\java" )
 else (
    set JAVA_EXE="%JAVA_HOME%\jre\bin\java" )
```

with the following code:

```
set JAVA_EXE="%JAVA_HOME%\jre\bin\java"
```

c. Remove all quotations from the following Java system properties:.

```
set CLIENTSOAP=-Dcom.ibm.SOAP.ConfigURL=file:%USER_INSTALL_
ROOT%\properties\soap.client.props
set CLIENTSAS=-Dcom.ibm.CORBA.ConfigURL=file:%USER_INSTALL_
ROOT%\properties\sas.client.props
set CLIENTSSL=-Dcom.ibm.SSL.ConfigURL=file:%USER_INSTALL_
```

```
ROOT%\properties\ssl.client.props
set CLIENTIPC=-Dcom.ibm.IPC.ConfigURL=file:%USER_INSTALL_
ROOT%\properties\ipc.client.props
```

d. Remove all trailing white space characters from the entire file.

4.2.2.3 Running wsadmin.sh or wsadmin.bat from the Command Line

Ensure that the script works by running `wsadmin.sh` or `wsadmin.bat` from the command line. Note the following:

- You may need to enter the user name and password at the login prompt.
- You may need to accept the server certificate by clicking **Y** at the signer exchange prompt.

4.2.2.4 Editing the sas.client.props File

See Step 20 of Section 4.2.4, "Creating an Application Server Connection" for instructions.

4.2.2.5 Creating an Application Server Connection in Oracle JDeveloper

Follow the instructions in Section 4.2.4, "Creating an Application Server Connection" to create an application server connection, and enter the following information when prompted:

- Use the `wsadmin.sh` or `wsadmin.bat` file you created in this section (for example, `/home/user/IBM/AppClient/wsadmin.sh`).
- Use the runtimes directory of the WebSphere Application Client (for example, `/home/user/IBM/AppClient/runtimes`).
- Use the properties directory that contains `sas.client.props` (for example, `/home/user/IBM/AppClient/properties`).

4.2.3 Configuring the Proxy on IBM WebSphere Server

1. Log in to the IBM WebSphere Administrative Console:

   host:*port*/ibm/console

2. Go to **Application servers** > **ServerName** > **Process definition** > **Java Virtual Machine** > **Custom properties**.
3. Define the following properties and values.

Property	Value
http.proxyHost	www-proxy.us.oracle.com
http.proxyPort	80
http.proxySet	true

4. Restart the server.

4.2.4 Creating an Application Server Connection

You must create a connection to the IBM WebSphere Server to which to deploy a SOA composite application. During application server connection creation, you are prompted for configuration information on several wizard pages. Table 4–2 describes

where to find this information on IBM WebSphere Administrative Console for which you are prompted. The locations differ based on the type of IBM WebSphere Server you are using, and the server where the application is being deployed.

Table 4–2 Location of Application Server Connection Configuration Details

Connection Wizard Fields	For IBM WebSphere Application Server - Network Deployment (ND), Select...	For IBM WebSphere Application Server 7.0, Select...
Configuration Page		
▪ **SOAP Connector Port**	**System administration > Deployment manager > Configuration > Ports > SOAP_CONNECTOR_ADDRESS**	**Servers > Server Types > WebSphere Application Servers > *Your_Server_Name* > Configuration > Ports > SOAP_CONNECTOR_ADDRESS**
▪ **Server Name**	**System administration > Deployment manager > Configuration > Name**	**Servers > Server Types > WebSphere Application Servers > *Your_Server_Name* > Configuration > Name**
▪ **Target Node**	**System administration > Deployment manager > Runtime > Node name**	**Servers > Server Types > WebSphere Application Servers > *Your_Server_Name* > Runtime > Node name**
▪ **Target Cell**	**System administration > Deployment manager > Runtime > Cell name**	**Servers > Server Types > WebSphere Application Servers > *Your_Server_Name* > Runtime > Cell name**
JMX Page		
▪ **RMI Port**	**System administration > Deployment manager -> Configuration > Ports > BOOTSTRAP_ADDRESS**	**Servers > Server Types > WebSphere Application Servers > *Your_Server_Name* > Configuration > Ports > BOOTSTRAP_ADDRESS**

Note: If you are using IBM WebSphere ND as the server type and you are deploying the application to the deployment manager server, then use the second column in the table to locate the configuration information you need.

To create an application server connection:

1. From the **File** main menu, select **New**.
2. In the **General** list, select **Connections**.
3. Select **Application Server Connection**, and click **OK**.

   The Name and Type page appears.
4. In the **Connection Name** field, enter a name for the connection.
5. In the **Connection Type** list, select **WebSphere Server 7.x** to create a connection to IBM WebSphere Server.
6. Click **Next**.

   The Authentication page appears.
7. In the **Username** field, enter the user authorized for access to the application server.
8. In the **Password** field, enter the password for this user.
9. Click **Next**.

The Configuration page appears. If you are not sure about the information to enter on this page, see Table 4–2.

10. In the **Host Name** field, enter the host on which the IBM WebSphere Server is installed. If no name is entered, the name defaults to `localhost`.
11. In the **SOAP Connector Port** field, enter the port number of the server on which IBM WebSphere Server is installed. The default SOAP connector port is `8879`.
12. In the **Server Name** field, enter the name assigned to the target application server for this application.
13. In the **Target Node** field, enter the name of the target node for this connection. A node is a grouping of managed servers (for example, *host*`Node01`, where *host* is the name of the host on which the node resides).
14. In the **Target Cell** field, enter the name of the target cell for this connection. A cell is a group of processes that host runtime components (for example, *host*`Node01Cell`, where *host* is the name of the host on which the node resides).
15. In the **Wsadmin script location** field, enter or browse for the location of the `wsadmin` script file to use for defining the system login configuration for this application server connection (for example, *WAS_HOME*`\bin\wsadmin.bat` for Windows or *WAS_HOME*`/bin/wsadmin.sh` for Unix).

    Note: Do not enter spaces in the path to the `wsadmin.sh` or `wsadmin.bat` file. For example, if on Windows, use the DOS equivalent path of `C:\Progra~1\` instead of `C:\Program Files\`.

16. Click **Next**.

    The JMX page appears.

17. If you want to browse the SOA Infrastructure and deploy over JMX, select **Enable JMX for this connection**.
18. In the **RMI Port** field, enter the port number for the IBM WebSphere Server's RMI connector port. If you are not sure about the information to enter on this page, see Table 4–2.
19. In the **WebSphere Runtime Jars Location** field, enter or browse for the IBM WebSphere Server's runtime JAR files (for example, *WAS_HOME*`/runtimes`).
20. In the **WebSphere Properties Location (for secure MBean access)** field, enter or browse for the location of the file that contains the properties for the security configuration and MBeans that are enabled (for example, *WAS_HOME*`/profiles/profile_name/properties`). This field is optional (for some Oracle JDeveloper use cases), but is required for SOA browsing and deployment. The location you specify must contain the `sas.client.props` file. Details about the contents of the `sas.client.props` file are as follows:

    - Authentication:

      The `sas.client.props` file is required for authentication, and must be edited as follows:

      ```
      com.ibm.CORBA.securityServerHost=Server_Host_Name
      com.ibm.CORBA.securityServerPort=RMI/BOOTSTRAP_Port

      com.ibm.CORBA.loginSource=properties
      ```

```
com.ibm.CORBA.loginUserid=User_Name
com.ibm.CORBA.loginPassword=Plain_Text_or_Encoded_Password
```

- Encode password:

  To encode the password in the `sas.client.props` file, save this file with a clear text password and then run the following utility:

  On Windows:

  ```
  WAS_HOME\bin\PropFilePasswordEncoder.bat
  ..\properties\sas.client.props com.ibm.CORBA.loginPassword
  ```

  On UNIX:

  ```
  WAS_HOME/bin/PropFilePasswordEncoder.sh
  ../properties/sas.client.props com.ibm.CORBA.loginPassword
  ```

- SSL (If not required):

  In most cases, SSL is not required for JMX. You must explicitly disable SSL as follows:

  ```
  # Does this client support/require SSL connections?
  com.ibm.CSI.performTransportAssocSSLTLSRequired=false
  com.ibm.CSI.performTransportAssocSSLTLSSupported=false
  ```

- SSL (If required):

  If you require SSL for JMX, do *not* configure `ssl.client.props`. Instead, you must append the necessary SSL configuration details to `sas.client.props` for Sun JRE clients, since Oracle JDeveloper runs in the Sun JRE.

  Edit the following two sections in `sas.client.props`:

  – Edit the section on SSL connection requirements.

  ```
  # Does this client support/require SSL connections?
  com.ibm.CSI.performTransportAssocSSLTLSRequired=false
  com.ibm.CSI.performTransportAssocSSLTLSSupported=true
  ```

  – Append the following syntax to the end of `sas.client.props`. For the `com.ibm.ssl.trustStore` property, you can use the path to any `*.jks` truststore.

  ```
  #-------------------------------------------------------------

  # SSL configuration alias referenced in ssl.client.props
  #-------------------------------------------------------------

  com.ibm.ssl.alias=JDeveloperSSLSettings
  com.ibm.ssl.protocol=SSL
  com.ibm.ssl.securityLevel=HIGH
  com.ibm.ssl.trustManager=SunX509
  com.ibm.ssl.keyManager=SunX509
  com.ibm.ssl.contextProvider=SunJSSE
  com.ibm.ssl.enableSignerExchangePrompt=gui

  com.ibm.ssl.trustStoreName=DemoTrustStore
  com.ibm.ssl.trustStore=c:/YOUR_JDEVHOME/your_server/
  lib/DemoTrust.jks

  com.ibm.ssl.trustStorePassword=DemoTrustKeyStorePassPhrase
  ```

```
com.ibm.ssl.trustStoreType=JKS
com.ibm.ssl.trustStoreProvider=SUN
com.ibm.ssl.trustStoreFileBased=true
com.ibm.ssl.trustStoreReadOnly=false
```

– Upon the first invocation of JMX (typically when you click **Test Connection** on the Test page of this wizard), the SSL Signer Exchange dialog can appear. Click **y** to accept the server certificate. Note that a `ThreadDeath` error is displayed that can safely be ignored.

– Provide the keystore location through the system properties in either of the following ways:

Note: When configuring the truststore location through the system properties on Windows operating systems, you must enter a forward slash (`/`) in the path. For example, `c:/to/path/truststore`.

From the command prompt:

```
$JDEV_INSTALL_DIR/jdev/bin/jdev
-J-Djavax.net.ssl.trustStore=c:/path/to/truststore
-J-Djavax.net.ssl.trustStorePassword=DemoTrustKeyStorePassPhrase
```

In the `jdev.conf` file:

```
AddVMOption   -Djavax.net.ssl.trustStore=c:/path/to/truststore
AddVMOption   -Djavax.net.ssl.trustStorePassword=DemoTrust
 KeyStorePassPhrase
```

- Multiple WAS connections

  Since one `sas.client.props` file is required for each application server connection, Oracle recommends that you create a directory for each application server, copy `sas.client.props` to that directory, and edit the file as necessary.

21. Click **Next**.
22. Click **Test Connection** to test your server connection.
23. If the connection is successful, click **Finish**. Otherwise, click **Back** to make corrections in the previous dialogs. Even if the connection test is unsuccessful, a connection is created.

4.2.5 Deploying SOA Composite Applications

Deployment of SOA Composite Applications from Oracle JDeveloper to IBM WebSphere Server is largely the same as described in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

The only exception is the appearance of the **Deploy using SSL** check box on the SOA Servers page of the deployment wizard. This differs from Oracle WebLogic Server, where the **Deploy using SSL** check box instead appears on the Configuration page of the Create Application Server Connection wizard page.

Table 4–3 describes what occurs when you select this check box during IBM WebSphere Server deployment.

Table 4–3 Deployment to HTTPS and HTTP Servers

If This Checkbox Is...	Then...
Selected	An HTTPS server URL must exist to deploy the composite with SSL. Otherwise, deployment fails. If the server has only an HTTP URL, deployment also fails. This enables you to ensure that SSL deployment must *not* go through a non-SSL HTTP URL, and must only go through an HTTPS URL
Not selected	An HTTP server URL must exist to deploy to a non-SSL environment. Otherwise, deployment fails. If the server has both HTTPS and HTTP URLs, deployment occurs through a non-SSL connection. This enables you to force a non-SSL deployment from Oracle JDeveloper, even though the server is SSL-enabled.

4.2.6 Using EJB Bindings

If a SOA composite application includes an EJB service, you must perform the following configuration procedures for the EJB service binding to work properly:

- Section 4.2.6.1, "EJB Service Binding"
- Section 4.2.6.2, "EJB Client"
- Section 4.2.6.3, "EJB Reference Binding"

4.2.6.1 EJB Service Binding

You must set up credentials for EJB JNDI binding before deploying a composite that contains an EJB service binding.

1. Create an entry for Oracle Platform Security Services (OPSS) (for example, with `SOA` as the name and `Deployer` as the key.
   a. Go to the *`MW_HOME`*`/oracle_common/common/bin` directory.

      where *`MW_HOME`* is the directory in which Oracle SOA Suite is installed.

   b. Make the `wsadmin.sh` file executable (if it is not already):

      ```
      chmod +x wsadmin.sh
      ```

   c. Execute the following command, and enter the password when prompted:

      ```
      ./wsadmin.sh -host localhost -port 8880 -conntype SOAP -user adminusername
      -lang jython
      ```

      The port number is the `SOAP_CONNECTOR_ADDRESS` of the host used to connect to the server for deployment. In the IBM Administrative Console, navigate to the Ports table via **Deployment Manager > Ports** to locate the value.

   d. Enter the following command to create the credentials:

      ```
      Opss.createCred(map="SOA",key="Deployer",user="adminusername",password="pas
      sword
      ")
      ```

2. Assign the JNDI reading, writing, and binding roles to the administrator user.

Note: The JNDI binding role does not need to be granted to the Administrator. However, it must match the user you specified with the `Opss` command in Step d.

a. Log in to the WebSphere Administrative Console.

b. Click and expand **Environment** > **Naming**.

c. Click **CORBA naming service groups**.

d. Click the **Add** button.

e. Select all the roles in the selection box at the top.

f. Search for groups using the wildcard ("`*`")

g. Select the **Administrators** group (to which the `adminusername` user belongs).

h. Click **OK**.

i. Click the **Save** link.

j. Restart the server.

4.2.6.2 EJB Client

Generate stubs for the EJB interfaces using the `createEJBStubs.sh` utility and ensure that the stubs are in the client classpath.

4.2.6.3 EJB Reference Binding

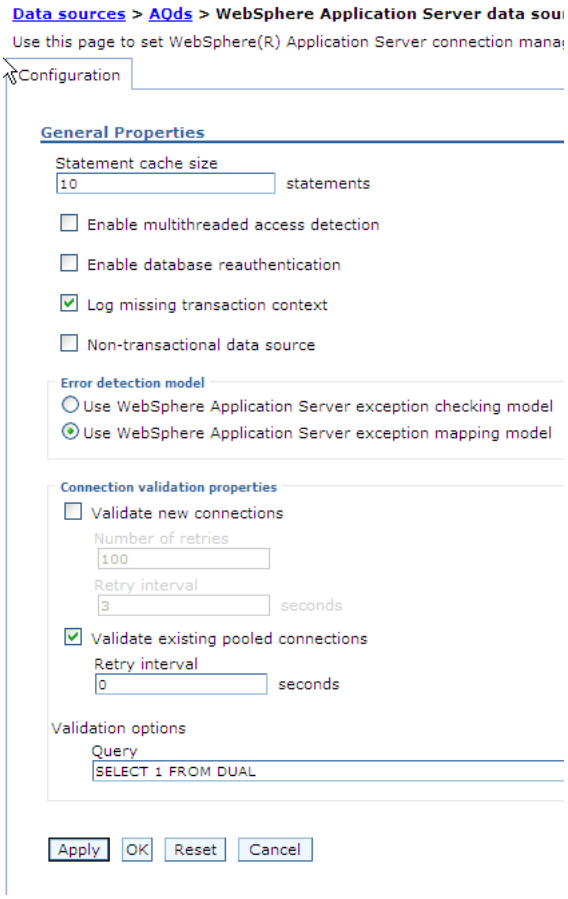
You must include the EJB stubs for the external EJB interface in the composite `SCA-INF/classes` or `SCA-INF/lib` directory.

4.2.7 AQ Technology Adapter and WebSphere 7.0

For the AQ Adapter to work correctly on the WebSphere 7.0 platform, you need to use the IBM WebSphere Administrative Console to provide specific connection factory and data source properties.

For the connection factory, you need to set the following custom property for the connection pool: `defaultConnectionTypeOverride = unshared`

For the AQ adapter dataSource, ensure that `validate existing pooled connections` is checked. The associated interval can be set to 0. See the following screen shot.



Also for the AQ adapter dataSource, you must define the same property as a custom property for the connection pool by setting the following:

```
defaultConnectionTypeOverride = unshared
```

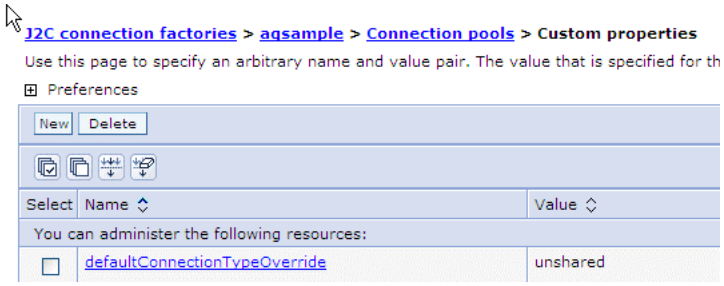
See the following screen shot.

You also need to set the maximum connections value of AQAdapter J2C connection factories to a higher value than the default of 10. You can find this entry in the WebSphere Application Server J2C connection factories -> <Name of AQAdapter> -> Connection pools panel.

4.2.8 JMS Technology Adapter on WebSphere 7.0

If you are developing composite applications to run on WebSphere 7.0, you need to use the Third Party option when modelling the JMS adapter with the Default Messaging JMS provider. You can specify that the adapter uses a third-party JMS Provider, by supplying a value to the FactoryProperties parameter in the `weblogic-ra.xml` file. Specifically, you can provide the `ThirdPartyJMSProvider` value to the FactoryProperties parameter.

When deployed on WebSphere 7.0, the JMS Adapter will not work with an AQJMS provider, unless you use the Adapter Configuration Wizard to set `defaultConnectionTypeOverride` as unshared for both the adapter connection factory pool and for the queue/topic connection factory pool. See the following screen shot.



You also need to set the maximum connections value of JMS Adapter J2C factories to a higher value than the default of 10. You can find this entry in the **WebSphere Application Server J2C connection factories** > *Name of JMS Adapter* > **Connection pools** panel.

4.2.8.1 Avoiding JMS Adapter Connection Leaks

While running JMS Adapter use cases on WebSphere 7.0, you might encounter the following error from a connection leak:.

```
java.lang.IllegalStateException: ConnectionManager is null
```

To avoid connection leaks, update the maximum and minimum connection value for the JMS Adapter to the same value at **Queue connection factories** > *Connection_factory_name* > **Connection pools and J2C connection factories** > *J2C_Connection_Factoryname* > **Connection Point**.

4.2.9 Oracle Database Adapter on WebSphere 7.0

For the Oracle Database Adapter to work properly, you need to the set the maximum connections value of the DB adapter J2C connection factories, using the WebSphere Admin Server. This value needs to be set to a higher value than the default of 10. You can find this entry under **J2C connection factories** > *Name of DB-Adapter* > **Connection pools**. The preferred value is 100.

4.3 Differences and Restrictions When Managing Oracle SOA Suite Components on IBM WebSphere

The following sections describe differences and restrictions when managing Oracle SOA Suite components on IBM WebSphere:

- Section 4.3.1, "Publishing Services to a UDDI Registry"

- Section 4.3.2, "Oracle Enterprise Manager Fusion Middleware Control Console Shortcut Links"

4.3.1 Publishing Services to a UDDI Registry

You cannot publish service binding components to the Universal Description, Discovery, and Integration (UDDI) registry from Oracle Enterprise Manager Fusion Middleware Control on IBM WebSphere.

4.3.2 Oracle Enterprise Manager Fusion Middleware Control Console Shortcut Links

Oracle Enterprise Manager Fusion Middleware Control does not include shortcut links to the WebSphere Administrative Console from the following locations:

- The **Server Data Source JNDI** and **Server Transaction Data Source JNDI** fields of the **Data Sources** section of the SOA Infrastructure Common Properties page
- The **Related Links** menu available on service engine pages.

To log in to IBM WebSphere, you must go directly to the WebSphere Administrative Console.

4.3.3 DefaultToDo Task Flow is Configured to Use HTTPS

Oracle SOA Suite on IBM WebSphere is configured to use HTTPS. This means the `DefaultToDo` task flow also uses HTTPS because the `DefaultToDo` task flow host name, port, and protocol are based on the SOA Server URL.

If a valid certificate is not available on the server, then `DefaultToDo` would not be accessible in Microsoft Internet Explorer and Google Chrome, while Mozilla Firefox would issue a warning and then allow the user to proceed. If necessary, use Oracle Enterprise Manager Fusion Middleware Control to change the SOA Server URL.

4.4 Differences and Restrictions When Managing Oracle BAM on IBM WebSphere

The following sections describe differences and restrictions when using Oracle BAM on IBM WebSphere:

- Section 4.4.1, "Configuring Oracle BAM Adapter"
- Section 4.4.2, "Using Oracle Data Integrator with Oracle BAM"
- Section 4.4.3, "Using ICommand"
- Section 4.4.4, "Configuring Logging for Oracle BAM on IBM WebSphere"
- Section 4.4.5, "Configuring Trusted Domains"
- Section 4.4.6, "Configuring Security"
- Section 4.4.7, "Using Oracle Internet Directory with Oracle BAM"
- Section 4.4.8, "Configuring Enterprise Message Sources to Connect to Remote JMS Queue/Topics"
- Section 4.4.9, "Using Oracle BAM Data Controls"
- Section 4.4.10, "Configuring the LTPA Timeout for Active Data Reports"

4.4.1 Configuring Oracle BAM Adapter

Configuration of Oracle BAM Adapter on IBM WebSphere is largely the same as described in *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*. The exception is that you use the IBM WebSphere Administrative Console (instead of the Oracle WebLogic Server Administration Console) to configure Oracle BAM Adapter.

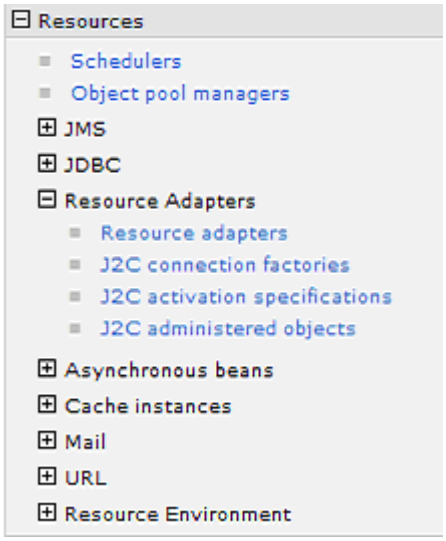
Refer to *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite* for complete information. The information provided in this section simply highlights the selections you make when using the IBM WebSphere Administrative Console to configure Oracle BAM Adapter properties, connection factories and trusted domains.

Note: When updating property values in the IBM WebSphere Administrative Console, click the property to open a page, enter the values as needed, and click **OK**. To commit the changes, click **Save**. Then restart Oracle SOA Server.

4.4.1.1 Configuring Oracle BAM Adapter Properties

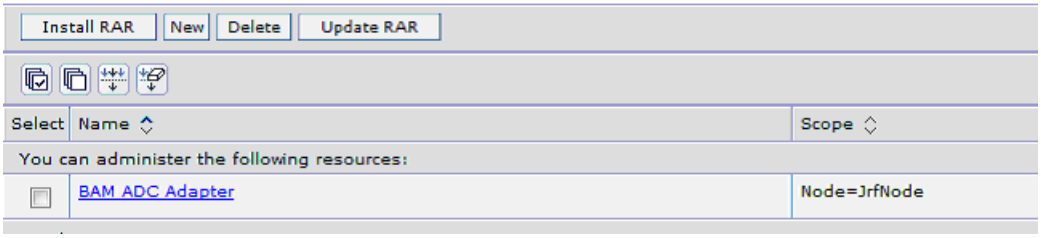
In the IBM WebSphere Administrative Console, you navigate to **Resources > Resource Adapters** (Figure 4–1) to locate the Oracle BAM Adapter resource.

Figure 4–1 Resources and Resource Adapters Panels in Administrative Console



In the Resource Adapter summary table, you click the Oracle BAM Adapter resource name to configure the properties (for example, **OracleBAMAdapter** or **BAM ADC Adapter** as shown Figure 4–2. The name varies depending on how it was deployed).

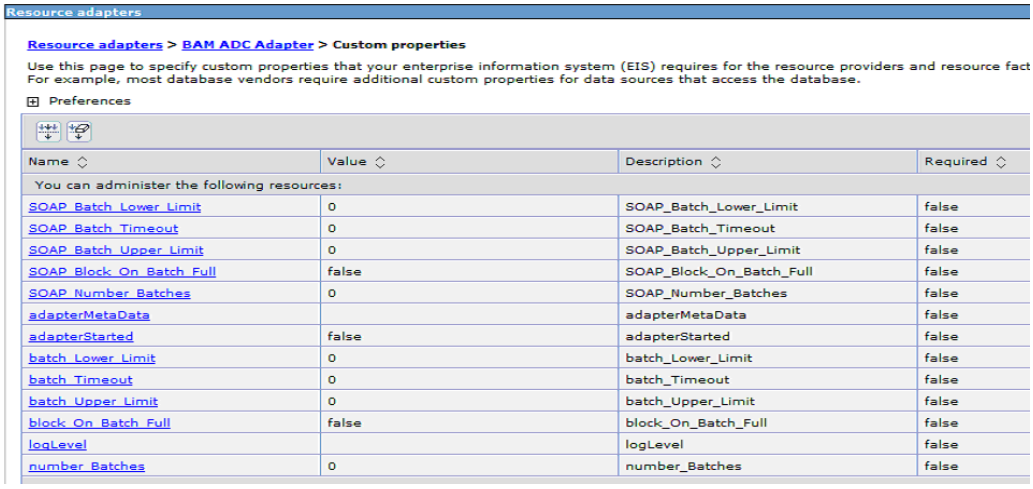
Figure 4–2 Resource Adapter Summary Table



On the Configuration page, you click **Custom properties** in the **Additional Properties** section on the right (Figure 4–3) to display all the properties you can configure for the selected Oracle BAM Adapter, as shown in Figure 4–4.

Figure 4–3 Additional Properties Section

Figure 4–4 Custom Properties Page of Oracle BAM Adapter



4.4.1.2 Configuring Oracle BAM Connection Factories

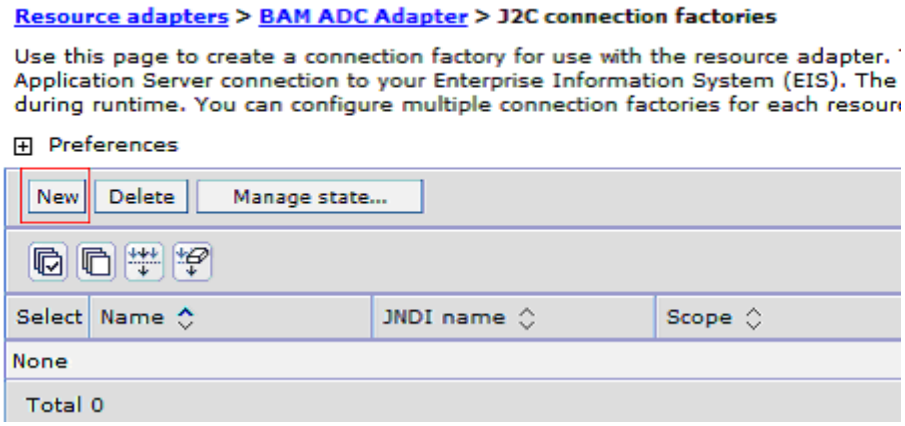
Before deploying applications that use Oracle BAM Adapter, a connection factory to Oracle BAM Server must be configured. You can configure both Remote Method Invocation (RMI) and Simple Object Access Protocol (SOAP) connection factories.

After clicking an Oracle BAM Adapter resource name as shown in Figure 4–2, on the Configuration page, you click **J2C connection factories** in the **Additional Properties** section on the right (Figure 4–5) to display a list of configured connection factories that you can use with the resource adapter.

Figure 4–5 Additional Properties Section

If there are no connection factories listed on the J2C Connection Factories page, click **New** to create and configure an Oracle BAM connection factory to Oracle BAM Server (Figure 4–6). You can create connection factories for RMI-based calls and SOAP-based calls.

Figure 4–6 J2C Connection Factories Page



When creating RMI-based and SOAP-based connection factories, provide a connection factory name, a JNDI name, and the Connection factory interface for each type (Figure 4–7 and Figure 4–8).

Figure 4–7 New J2C Connection Factory Configuration

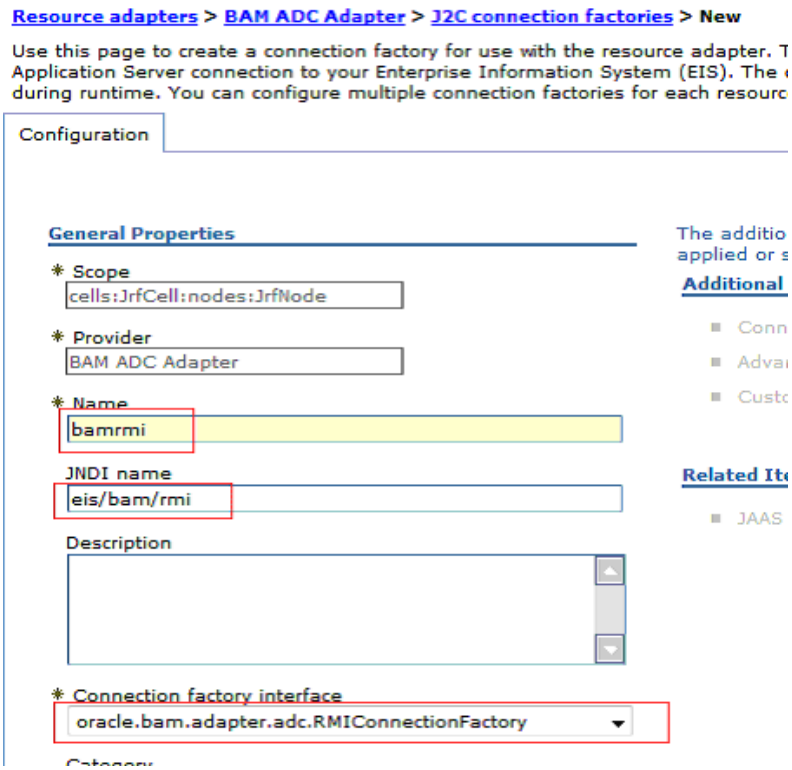


Figure 4–8 SOAP Connection Factory Configuration

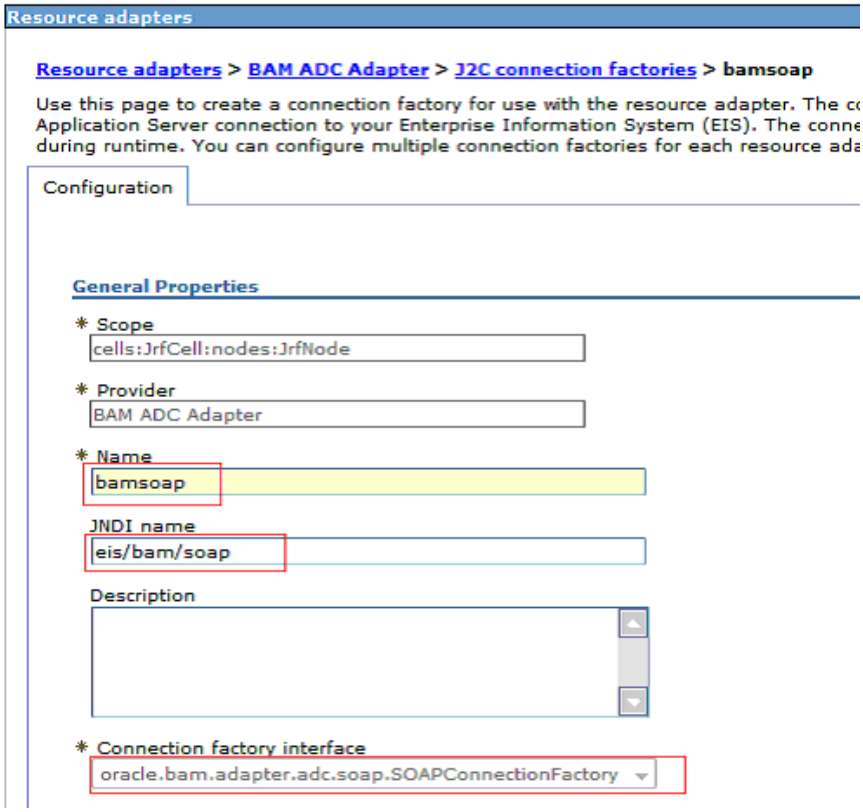
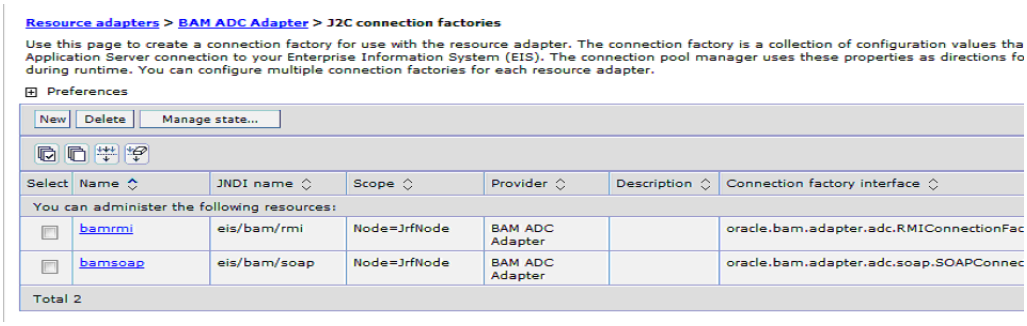


Figure 4–9 shows the J2C Connection Factories page with two connection factories created and listed in the table. Note that the node and cell names will vary depending on the deployment.

Figure 4–9 J2C Connection Factories Page



To configure the properties for a connection factory, click the connection factory name (for example, **bamrmi** or **bamsoap**), then on the Configuration page click **Custom properties** on the right. Figure 4–10 and Figure 4–11 show the custom properties you can configure for a RMI-based connection factory and a SOAP-based connection factory, respectively. Note that with RMI-based connection factories, **InstanceName** is the connection name for Oracle BAM Adapter (for example, `ADCAdapter1`), and **PortNumber** is the `BOOTSTRAP_ADDRESS` of the Oracle BAM Server. With SOAP-base connection factories, **PortNumber** is the `WC_defaulthost` of Oracle BAM Server.

Figure 4–10 Connection Factory Custom Properties for RMI-Based Calls

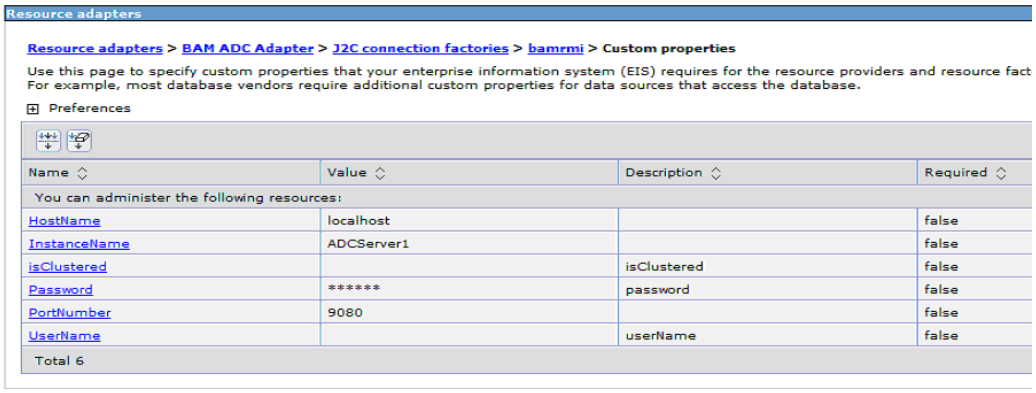


Figure 4–11 Connection Factory Custom Properties for SOAP-Based Calls

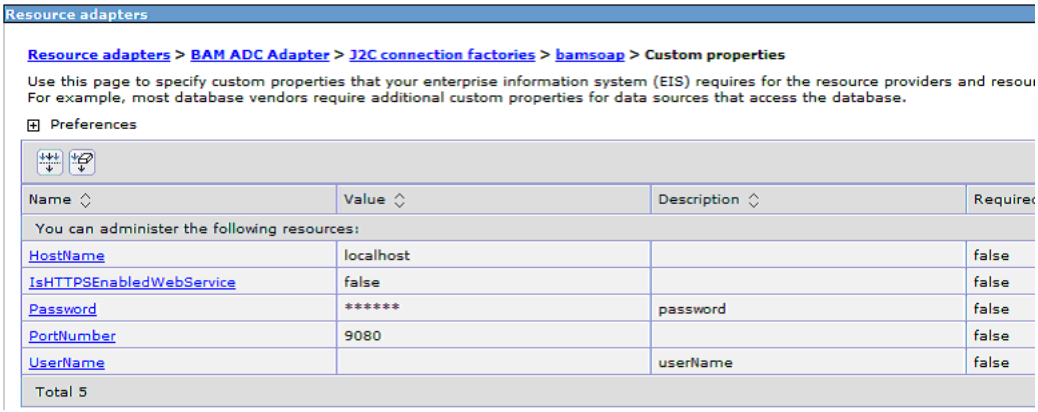


Figure 4–11 also shows a SOAP-based connection factory configured for HTTP. To configure an HTTPS SOAP-based connection factory, create a new connection factory and specify the `IsHTTPSEnabledWebService` property value as `true`.

4.4.1.3 Configuring Trusted Domains

When using the RMI connection between a SOA composite application and Oracle BAM Server, that is when they are deployed in different cells, trusted domain configuration must be done in the IBM WebLogic Administrative Console. For more information, see Section 4.4.5, "Configuring Trusted Domains."

4.4.2 Using Oracle Data Integrator with Oracle BAM

Setting up the Oracle BAM and Oracle Data Integrator integration with Oracle BAM Server running on IBM WebSphere is largely the same as described in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*, with a few exceptions. The exceptions are:

1. If you already have an installation of Oracle Data Integrator 10g working with an older version of Oracle BAM, you must have another installation of Oracle Data Integrator 10g to work with the current release of Oracle BAM. You cannot use the same Oracle Data Integrator 10g installation to work with multiple versions of Oracle BAM.
2. Apache Ant is required to run the installation script. Set the environment variable `ANT_HOME` to the location where ANT is installed before you run the `bam_odi_configuration.sh` (`bam_odi_configuration.bat`) script.
3. Set the following environment variables before you run the installation script:
   - `JAVA_HOME`: Root directory of the supported version of Java Development Kit (see the Oracle BAM support matrix on Oracle Technology Network web site for supported JDK versions).
   - `WAS_HOME`: The location of the IBM WebSphere Application Server installation directory.
   - `WAS_CLIENT_PROPS`: Directory where the `sas.client.props` file that the user wants to use resides.
4. Before you run the installation script, make sure login security values in `sas.client.props` and the server port value in `BAMICommandConfig.xml` are configured properly. For information, see Section 4.4.3, "Using ICommand."

5. After running the installation script and before using Oracle Data Integrator with Oracle BAM Server running on IBM WebSphere, make sure the server port value in `BAMODIConfig.xml` is configured to the same server port value as in step 4 above. To change the value, locate `BAMODIConfig.xml` in `$ODI_HOME/oracledi/lib/config`, then uncomment the line for the server port value.

4.4.3 Using ICommand

When a standalone Oracle BAM client (such as ICommand, Oracle Data Integrator, and Oracle BAM Data Control) connects to Oracle BAM Server, the configuration file (for example `BAMICommandConfig.xml`), which is read when the Oracle BAM client code is invoked, must point to the server on which the Oracle BAM Server instance is running.

In addition, login security must be configured before standalone Oracle BAM clients can connect to Oracle BAM Server.

4.4.3.1 Configuring Oracle BAM Server Port

By default ICommand looks for Oracle BAM Server on port 2809. If the Oracle BAM Server port number is changed from the default during the setup and configuration of Oracle BAM on IBM WebSphere, then you must manually change the port number from 2809 to the new port number in the `BAMICommandConfig.xml` file.

Locate the `BAMICommandConfig.xml` file in *`SOA_ORACLE_HOME`*`/bam/config`.

The property to change is:

```
<ServerPort>2809</ServerPort>
```

To determine the correct port value to use:

- On IBM WebSphere ND: Use the IBM WebSphere Administrative Console to navigate to **Servers > Server Types > WebSphere Application Servers > [bam_server_name] > Ports** to locate the `BOOTSTRAP_ADDRESS` value of the Oracle BAM Server.
- On IBM WebSphere AS: Look at the `BOOTSTRAP_ADDRESS` value in the file `portdef.props`, which is located in `WAS_HOME/was_profiles/DefaultTopology/was_as/ServerName/properties`.

The `BAMICommandConfig.xml` file should also have the following `ServerPlatform` property:

```
<ServerPlatform>websphere</ServerPlatform>
```

4.4.3.2 Configuring Login Security

For information, see Section 4.4.6.1, "Configuring Login Security for Standalone Oracle BAM Components on IBM WebSphere."

4.4.4 Configuring Logging for Oracle BAM on IBM WebSphere

To configure logging for Oracle BAM on IBM WebSphere, you have to use either the IBM WebSphere Administrative Console or execute `wsadmin` scripts.

To use the IBM WebSphere Administrative Console to configure logging:

1. Log in to the IBM WebSphere Administrative Console.

2. In the navigation panel, expand **Servers > Server Types**.
3. Select **WebSphere application servers**, then select the server that is hosting the Oracle BAM application (for example, **bam-server1** on IBM WebSphere ND or **ServerName** on IBM WebSphere AS).
4. On the **Configuration** tab, **Troubleshooting** section, select **Change Log Detail Levels**, then expand **[All Components]**.

   You will see a list of known loggers.
5. Scroll down and select the desired **oracle.bam** logger.
6. Select **Message And Trace Levels** and set the desired level.

   You can set levels at any point in the package hierarchy right down to the individual class. This mechanism is analogous to modifying the `logging.xml` file.
7. Click **Apply** or **OK**, then click **Save to the master configuration**.

   This saves the changes permanently so they are in effect even if you restart IBM WebSphere.

   The log files are located at `WAS_HOME/was_profiles/DefaultTopology/was_as/ServerName/logs/ServerName`, (for example, `ServerName-diagnostic.log`), where `ServerName` is the name of the server that is hosting Oracle BAM.

Alternatively, you can execute `wsadmin` scripts to set the level for all the current descendants of a logger. For example:

```
wsadmin> myLoggers = OracleODL.listLoggers(pattern="oracle.bam.common.*")
wsadmin> for loggerName in myLoggers.keys():
wsadmin>   OracleODL.setLogLevel(target="ServerName", logger=loggerName,
level="FINE")
```

4.4.5 Configuring Trusted Domains

When Oracle BAM Server components require a connection to a remote server, trusted domain configuration must be done in the IBM WebSphere Administrative Console. For example, when Enterprise Message Sources (EMS) in Oracle BAM needs to connect to a topic/queue on a JMS server that is installed on a different IBM WebSphere instance, you have to set up the domain trust between the IBM WebSphere instances.

To perform communication with another server, IBM WebSphere has to retrieve a signer certificate from a secure remote SSL port during the handshake. The signer exchange process for setting up SSL to external servers such as Lightweight Directory Access Protocol (LDAP) is greatly simplified on IBM WebSphere. Instead of manually obtaining the remote server's signer certificate and then importing it into the appropriate trust store each time, the signer certificate retrieved from the remote port can be stored in an existing local trust store. Oracle BAM Server components that require a connection to the remote server can then use the validated signer certificate from the keystore.

To configure a trusted domain by obtaining and validating a signer certificate from a remote port:

1. Log in to the IBM WebSphere Administrative Console.

2. In the navigation panel, expand **Security**, then click **SSL certificate and key management**.
3. Click **Key stores and certificate**s.

   The **Keystore usages** dropdown should show **SSL keystores** as the value.
4. Select a trust store (for example, **NodeDefaultTrustStore**).
5. Click **Signer certificates**, then click **Retrieve from port**.

   This option opens an SSL connection to retrieve the certificate.
6. Enter the host name of the machine on which the signer resides.
7. Enter the SSL port on the host machine.
8. Enter an alias.
9. Click **Retrieve signer information**.
10. Verify the signer certificate information and the SHA digest of the certificate, which is used to ensure the information has not been modified in transit.
11. Click **Apply** or **OK** to add the signer certificate to the selected trust store.

4.4.6 Configuring Security

Login security must be configured before standalone Oracle BAM clients (such as Oracle Data Integrator, Oracle BAM Data Control and ICommand) can connect to Oracle BAM Server on IBM WebSphere.

Oracle BAM web applications by default use FORM as the authentication security method. To use the CLIENT_CERT authentication security method on IBM WebSphere, you must configure it manually.

To provide secure access to Oracle BAM web applications on IBM WebSphere, you must assign users to roles that provide the necessary permissions.

See the following for more information:

- Section 4.4.6.1, "Configuring Login Security for Standalone Oracle BAM Components on IBM WebSphere"
- Section 4.4.6.2, "Configuring Oracle BAM to Use CLIENT_CERT Authentication on IBM WebSphere"
- Section 4.4.6.3, "Creating User/Group Mappings for Oracle BAM on IBM WebSphere"

4.4.6.1 Configuring Login Security for Standalone Oracle BAM Components on IBM WebSphere

For standalone clients like Oracle Data Integrator, Oracle BAM Data Control and ICommand to connect to Oracle BAM Server on IBM WebSphere, certain property values must be set in the `sas.client.props` file, which is required for initial authentication of the standalone client by IBM WebSphere.

Edit the `sas.client.props` file to include the following properties:

```
com.ibm.CORBA.securityEnabled=true
com.ibm.CORBA.loginSource=properties
com.ibm.CORBA.securityServerHost=localhost
com.ibm.CORBA.securityServerPort=2809
com.ibm.CORBA.loginUserid=username
com.ibm.CORBA.loginPassword=password
```

```
com.ibm.CSI.performTransportAssocSSLTLSRequired=false
com.ibm.CSI.performTransportAssocSSLTLSSupported=false
```

...where `securityServerPort` is the deployment manager server `BOOTSTRAP_ADDRESS` value.

Using `WAS_HOME` as the root folder for the IBM WebSphere installation, the location of the `sas.client.props` file is:

- *`WAS_HOME`*`/profiles/<deployment_manager_profile_name>/properties` on IBM WebSphere ND
- *`WAS_HOME`*`/was_profiles/DefaultTopology/was_as/ServerName/properties` on IBM WebSphere AS

Details about the properties to configure in `sas.client.props` are found in Table 4–4, " Login Security Properties for the sas.client.props File".

Table 4–4 Login Security Properties for the sas.client.props File

Property to add...	Value to use...	Additional note about the property...
`com.ibm.CORBA.securityEnabled`	true	Must be set to this value
`com.ibm.CORBA.loginSource`	properties	Must be set to this value
`com.ibm.CORBA.securityServerHost`	<hostname>	Use localhost or the host name
`com.ibm.CORBA.securityServerPort`	<serverport>	Default port is 2809 The correct value can be determined by looking at the `BOOTSTRAP_ADDRESS` value: ■ Use the IBM WebSphere Administrative Console to navigate to **Servers > Server Types > WebSphere Application Servers > [bam_server_name] > Ports** to locate the `BOOTSTRAP_ADDRESS` value of the Oracle BAM Server on IBM WebSphere ND ■ Look at the `BOOTSTRAP_ADDRESS` value in the file `portdef.props`, which is located in *`WAS_HOME`*`/was_profiles/DefaultTopology/was_as/ServerName/properties` on IBM WebSphere AS
`com.ibm.CORBA.loginUserid`	<userid>	For example, `adminusername`

Table 4–4 (Cont.) Login Security Properties for the sas.client.props File

Property to add...	Value to use...	Additional note about the property...
com.ibm.CORBA.loginPassword	<password>	For example, password1 The loginPassword needs to be encrypted using the PropFilePasswordEncoder utility. The command to encrypt a password is: *WAS_HOME*/bin/PropFilePasswordEncoder.sh <path>/sas.client.props -SAS where <path> of the sas.client.props file is: ■ *WAS_HOME*/profiles/<deployment_manager_profile_name>/properties on IBM WebSphere ND ■ *WAS_HOME*/was_profiles/DefaultTopology/was_as/ServerName/properties on IBM WebSphere AS Instructions on how to use the utility are also provided in sas.client.props.
com.ibm.CSI.performTransportAssocSSLTLSRequired	false	SSL is not required
com.ibm.CSI.performTransportAssocSSLTLSSupported	false	SSL is not required

4.4.6.2 Configuring Oracle BAM to Use CLIENT_CERT Authentication on IBM WebSphere

On IBM WebSphere, Oracle BAM web applications must use FORM as the authentication security method and Oracle BAM web services must use BASIC as the authentication security method. Unlike Oracle WebLogic Server, IBM WebSphere does not provide a fallback mechanism for authentication methods, which means you cannot specify more than one authentication method. If you wish to use the CLIENT_CERT authentication security method for Oracle BAM web applications, you must configure it manually by following these steps:

1. Extract the existing oracle-bam-was.ear, located in *MW_HOME*/Oracle_SOA1/bam/applications, for example.
2. Modify the deployment descriptor web.xml in bam-web.war by replacing "FORM" with "CLIENT_CERT". For example:

```
<login-config>
  <auth-method>CLIENT-CERT</auth-method>
</login-config>
```

3. Repackage bam-web.war with the edited deployment descriptor.
4. Deploy the modified oracle-bam-was.ear.

4.4.6.3 Creating User/Group Mappings for Oracle BAM on IBM WebSphere

After installing Oracle BAM on IBM WebSphere AS or IBM WebSphere ND, you must specify the users and groups that are mapped to the security roles for Oracle BAM.

To create user/group mappings for Oracle BAM on IBM WebSphere:

1. Log in to the IBM WebSphere Administrative Console:

   host:port/ibm/console

2. On IBM WebSphere ND, navigate to the **Console Preferences** page in **System administration**. Select **Synchronize changes with Nodes** and click **Apply**.

   This ensures that all changes saved to the master configuration are propagated across the nodes.

3. In the navigation panel, expand **Applications > Application Types**.
4. Select **WebSphere enterprise applications**, then select **oracle-bam**.
5. On the **Configuration** tab, **Detail Properties** section, select **Security role to user/group mapping**.
6. Select the **bamuser** checkbox, then click **Map Users**.
7. Click **Search** to display a list of available users.
8. Select **cn=adminusername,dc=com** and move it to the **Selected** list, then click **OK** twice.
9. Save the change and restart Oracle BAM Server.

Alternatively, you can use the `wsadmin` command-line utility to configure the mapping. For example:

```
wsadmin> AdminApp.edit('oracle-bam','[-MapRolesToUsers
  ["bamuser" "No" "Yes" "cn=OracleSystemUser,dc=com" """bamuser" "No" "Yes"
"cn=adminusername,dc=com"""]')
wsadmin> AdminConfig.save()
```

4.4.7 Using Oracle Internet Directory with Oracle BAM

Using Oracle Internet Directory with Oracle BAM on IBM WebSphere is largely the same as described in *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*. The user `OracleSystemUser` must exist in the LDAP server. In addition, you must create user/group mappings for Oracle BAM on IBM WebSphere.

For instructions, see Section 4.4.6.3, "Creating User/Group Mappings for Oracle BAM on IBM WebSphere."

4.4.8 Configuring Enterprise Message Sources to Connect to Remote JMS Queue/Topics

For Enterprise Message Sources (EMS) on Oracle BAM Server to look up JMS resources hosted on a remote provider, you must first set up the trust between the local IBM WebSphere server (where Oracle BAM is deployed) and the remote IBM WebSphere server (where the JMS provider is configured). Then you set up the JMS resource on the remote server by creating a service integration bus, a JMS topic connection factory, and a JMS topic.

To connect to a remote JMS queue/topic from EMS:

1. Set up the trust between the remote IBM WebSphere instance and the local IBM WebSphere instance. For instructions, see Section 4.4.5, "Configuring Trusted Domains."
2. On the remote IBM WebSphere instance, log in to the IBM WebSphere Administrative Console.
3. To create a service integration bus, follow these steps:

a. In the navigator panel, expand **Service integration**. Click **Buses**, then click **New**.

b. Enter a name for your new bus (for example, `MyBus`).

   Note that this name should be different from the bus name in your local IBM WebSphere instance.

c. Deselect **Bus security**.

d. Click **Next**, then click **Finish**.

e. On the **Buses** page, click the bus name you just created.

f. On the Configuration tab, **Topology** section, click **Bus members** then click **Add**.

g. Choose the server to add to the bus from the dropdown list (for example, **JrfNode:JrfServer**).

h. Click **Next**, accepting all default values until you get to the Summary page, then click **Finish**.

4. To create a JMS topic connection factory, follow these steps:

   a. In the navigation panel, expand **Resources > JMS**.

   b. Click **Topic connection factories**.

   c. Expand Scope, then select the node and server as the scope from the dropdown list (for example, **Node=JrfNode,Server=JrfServer**).

      The scope identifies the level to which the resource (JMS topic connection factory) is visible.

   d. Click **New**, then select Default messaging provider as the provider that supports the topic connection factory instance, and click **OK**.

   e. In the **Administration** section of the Configuration page, enter a display name for the resource (for example, `myNewTopicCF`) and the JNDI name for the resource (for example, `jms/myNewTopicCF`).

   f. In the **Connection** section, from the **Bus name** dropdown list, select the bus to connect to (for example, **MyBus**).

      This is the service integration bus that the connection factory is used to create connections to.

   g. Enter the name of the target that is used to determine the messaging engine (for example, `JrfNode.JrfServer`).

      This is the bus member (server) you added in step 3g above.

   h. Select **Bus member name** as the type from the **Target Type** dropdown list.

   i. In the **Provider endpoints** box, enter `<yourhostname>:7277:` as the endpoint used to connect to a bootstrap server, then click **OK**.

5. To create a JMS topic, follow these steps:

   a. In the navigator panel, expand **Resources > JMS**.

   b. Click **Topics**.

   c. Expand **Scope**, then select the node and server as the scope from the dropdown list (for example, **Node=JrfNode,Server=JrfServer**).

d. Click **New**, then select **Default messaging provider** as the provider that supports the topic destination instance, and click **OK**.

e. In the **Administration** section of the Configuration page, enter a display name for the resource (for example, `myNewTopic`) and the JNDI name for the resource (for example, `jms/myNewTopic`).

f. In the **Connection** section, from the Bus name dropdown list, select the bus hosting the topic (for example, **MyBus**).

g. From the **Topic space** dropdown list, select **Create Service Integration Bus destination**.

h. Enter a name for the topic space and click **Next**, then click **Finish**.

The topic space name you created should now be listed in the **Topic space** dropdown list.

i. Click **OK**.

6. Save to the master configuration. Restart the server.

7. In Oracle BAM Architect on the local IBM WebSphere instance, create a new EMS definition using the remote provider URL, the remote connection factory (for example, `jms/myNewTopicCF`) and the remote topic (for example, `jms/MyNewTopic`) you created.

4.4.9 Using Oracle BAM Data Controls

Creating and using Oracle BAM data controls in Oracle JDeveloper is largely the same as described in *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*. Note, however, the exceptions described in Section 4.4.9.1, "Exceptions in JDeveloper."

When deploying an Oracle ADF application that uses Oracle BAM data controls, make sure you deploy the application to an IBM WebSphere application server where ADF shared libraries are available. Before deploying, the properties of the application server connection to IBM WebSphere created in JDeveloper must include the parameters as described in Section 4.4.9.2, "Application Server Connection Parameters."

4.4.9.1 Exceptions in JDeveloper

A few exceptions must be noted before using Oracle BAM data controls in JDeveloper. They are:

1. Copy the JAR files in Table 4–5 from IBM WebSphere to the following Oracle JDeveloper directory:

   JDEV_HOME/jdeveloper/was

Table 4–5 IBM WebSphere JAR Files to Copy and their Locations

JAR File to Copy	Location of JAR File on IBM WebSphere
com.ibm.ws.admin.client_7.0.0.0.jar	*WAS_HOME*/runtimes
com.ibm.ws.ejb.thinclient_7.0.0.jar	*WAS_HOME*/runtimes
com.ibm.ws.jpa.thinclient_7.0.0.jar	*WAS_HOME*/runtimes
com.ibm.ws.orb_7.0.0.jar	*WAS_HOME*/runtimes
ejb3exceptions.jar	*WAS_HOME*/runtimes
ibmorb.jar	*WAS_HOME*/java/jre/lib

Table 4–5 (Cont.) IBM WebSphere JAR Files to Copy and their Locations

JAR File to Copy	Location of JAR File on IBM WebSphere
oracle.webservices.standalone.client.jar	*MW_HOME*/oracle_common/modules/oracle.webservices_11.1.1
tools.jar	*WAS_HOME*/java/lib
wsclient_extended.jar	*MW_HOME*/oracle_common/webservices

2. Add the BAMCommonConfig.xml file to *JDEV_HOME*/jdeveloper/jdev/extensions/oracle.bam.jar.

   Note that oracle.bam.jar is available only after you have installed soa-jdev-extension.zip.

   BAMCommonConfig.xml should be added to the config directory in the root directory of the JAR file.

   The BAMCommonConfig.xml file should contain the following properties:

```
<ServerPlatform>websphere</ServerPlatform>
<ServerName>HOSTNAME</ServerName>
<ServerPort>BAMSERVERBOOTSTRAPADDRESS</ServerPort>
```

   For example:

```
<ServerPlatform>websphere</ServerPlatform>
<ServerName>myserver</ServerName>
<ServerPort>2801</ServerPort>
```

4.4.9.2 Application Server Connection Parameters

At runtime, Oracle BAM data controls in an Oracle ADF application use the Oracle BAM connection to connect to Oracle BAM Server on IBM WebSphere. Deploying an Oracle ADF application to IBM WebSphere is largely the same as deploying an ADF application to Oracle WebLogic Server. Note, however, that you must deploy the application to an IBM WebSphere application server where ADF shared libraries are available (for example, OracleAdminServer on IBM WebSphere ND). To enable this, certain parameters must be correctly set in the JDeveloper deployment profile for the application.

When you create the application server connection to IBM WebSphere in JDeveloper, on the Configuration page of the Create Application Server Connection wizard, make sure the parameters are properly set as shown in Table 4–6.

Table 4–6 Configuration Parameters for Server Connection

Parameter	Description
SOAP Connector Port	Port number of the host used to connect to the server for deployment, as defined in <SOAP_CONNECTOR_ADDRESS> on the IBM WebSphere Administrative Console.
Server Name	Name of server (as defined in IBM WebSphere) where the application is deployed.
Target Node	Name of the node (as defined in IBM WebSphere) where the application is deployed.
Target Cell	Name of the cell (as defined in IBM WebSphere) where the application is deployed.

Then on the JMX page of the Create Application Server Connection wizard, make sure the RMI port parameter is properly set as shown in Table 4–7.

Table 4–7 JMX Parameters for Server Connection

Parameter	Description
RMI Port	Port number of the IBM WebSphere application server's RMI connector port, as defined in <BOOTSTRAP_ADDRESS> on the IBM WebSphere Administrative Console.

Note: In the IBM WebSphere Administrative Console, the locations where you can find the values of <SOAP_CONNECTOR_ADDRESS> and <BOOTSTRAP_ADDRESS>, and the runtime node and cell names, differ based on the type of IBM WebSphere Server you are using and the server where the application is being deployed (for example, soa_server1 or the deployment manager server dmgr). For more information, see Table 4–2, " Location of Application Server Connection Configuration Details", which describes where to find the information in the IBM WebSphere Administrative Console.

4.4.10 Configuring the LTPA Timeout for Active Data Reports

On IBM WebSphere, the Lightweight Third Party Authentication (LTPA) timeout value specifies the period of time during which the server credentials from another server are valid. After the timeout period expires, the server credential from the other server must be revalidated.

The default LTPA timeout value is 120 minutes, which means the user is logged out after 120 minutes. The LTPA token and associated sessions are terminated and reauthentication is needed. This would affect, for example, users who have Oracle BAM applications and active data reports open in the browser for longer than 120 minutes.

To allow users to remain logged in for more than 120 minutes without having to log in again to reauthenticate credentials, set the LTPA timeout value to a higher number.

To change the LTPA timeout value:

1. Log in to the IBM WebSphere Administrative Console.
2. In the navigation panel, expand **Security** and click **Global Security**.
3. In the **Authentication** section on the right, click **LTPA**.
4. In the **LTPA timeout** field, enter a value in minutes.

   For example, to allow users to remain logged in for two days, enter 2880 minutes.

5

Managing Oracle WebCenter Portal on IBM WebSphere

This chapter contains information about installing, building, and managing Oracle WebCenter Portal applications and components on IBM WebSphere.

This chapter contains the following sections:

- Overview - Roadmaps
- Differences Installing and Configuring WebCenter Portal on IBM WebSphere
- Differences Developing and Deploying WebCenter Portal Applications on IBM WebSphere
- Differences Managing WebCenter Portal Components on IBM WebSphere
- Restrictions Using WebCenter Portal on WebSphere
- Troubleshooting WebCenter Portal on WebSphere

5.1 Overview - Roadmaps

The roadmaps in this section provide an overview of the steps required to install and configure WebCenter Portal on IBM WebSphere and point you to more detailed documentation. The steps required depend on whether you want to use WebCenter Portal: Spaces or build your own WebCenter Portal applications using WebCenter Portal: Framework. For details, see:

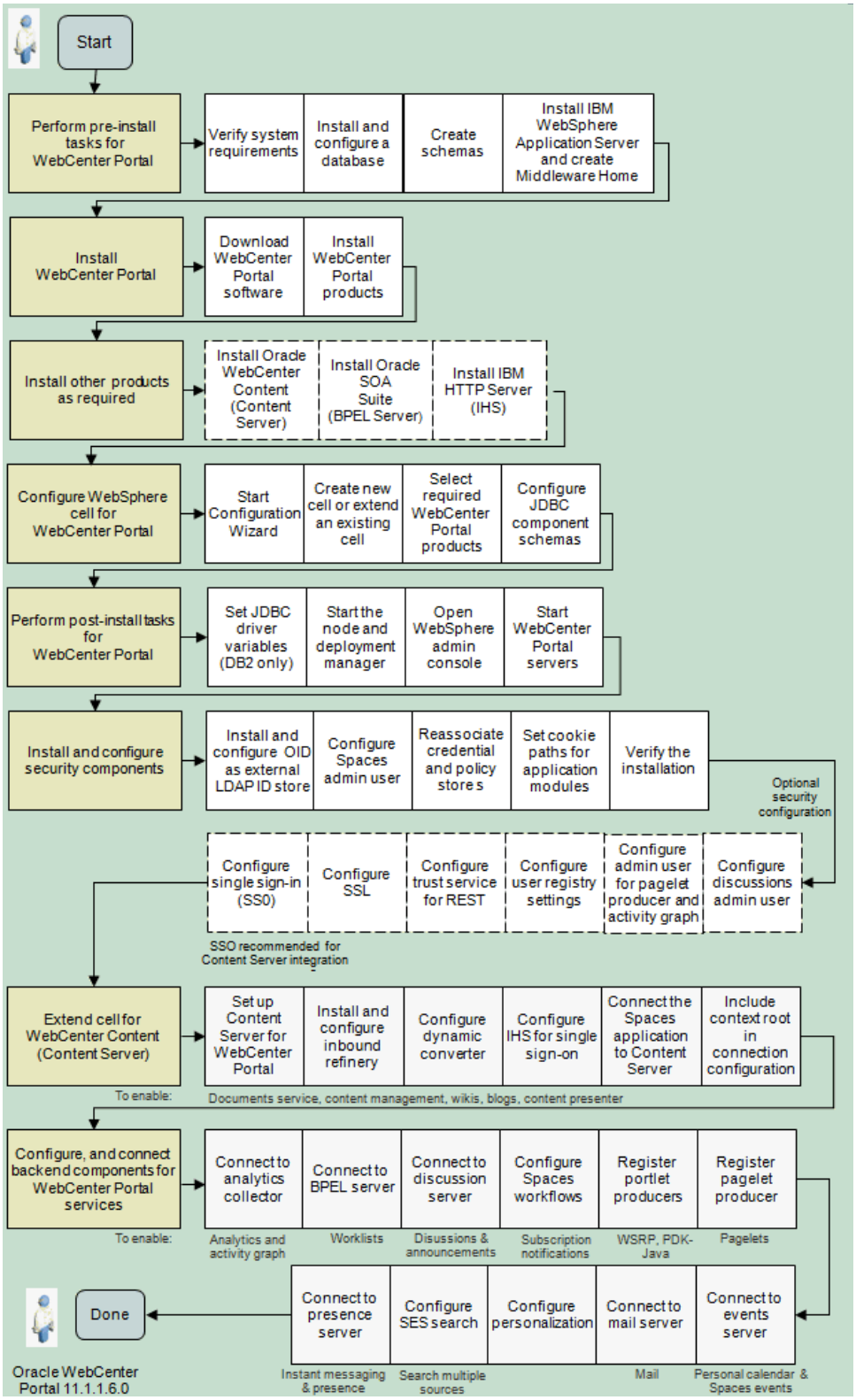
- Getting the Spaces Application Up and Running on IBM WebSphere
- Creating a WebSphere Cell for Framework Application Deployments

Click the flow charts for more information on how to complete each step.

5.1.1 Getting the Spaces Application Up and Running on IBM WebSphere

Figure 5–1 illustrates the installation and configuration process for WebCenter Portal: Spaces in a simple, non-clustered environment.

Figure 5–1 Getting the Spaces Application Up and Running on IBM WebSphere



Note: For deployment in a clustered environment, see Section 3.4, "Configuring Oracle Fusion Middleware High Availability on IBM WebSphere" and Section 5.2.19, "Configuring WebCenter Portal Applications for High Availability on IBM WebSphere".

Click the flow chart or use Table 5–1 to navigate to the appropriate documentation.

Table 5–1 Getting the Spaces Application Up and Running on IBM WebSphere - Simple Topology

Task and link to more information	Mandatory or Optional?	Notes
Verify system requirements	Mandatory	
Install and configure a database	Mandatory	
Create schemas for the Spaces application	Mandatory	
Install IBM WebSphere Application Server and create Middleware Home	Mandatory	
Install Oracle WebCenter Portal	Mandatory	
Install other products as required: ■ Oracle WebCenter Content ■ Oracle SOA Suite ■ IBM HTTP Server (IHS)	Optional	Oracle WebCenter Content is mandatory for content presenter, wikis and blogs, and recommended for the Documents service and the Spaces application. SOA is mandatory for the Worklist service and Spaces workflows. IHS is recommended for Oracle WebCenter Content Server integration and for single sign-on (SSO) since SSO is needed to stop multiple login prompts), and is required for REST and SOA.
Create new WebSphere cell for WebCenter Portal: Spaces	Mandatory	
Perform general post-install tasks for WebCenter Portal: ■ Set JDBC driver variables (DB2 only) ■ Start the node and deployment manager ■ Open WebSphere Admin console ■ Start WebCenter Portal servers	Mandatory	
Install and configure mandatory security components: ■ Install and configure Oracle Internet Directory (OID) as an external LDAP ID store ■ Configure admin user for the Spaces application ■ Reassociate credential and policy stores ■ Set cookie paths for Spaces application modules ■ Verify the WebCenter Portal installation	Mandatory	An external LDAP server is mandatory on IBM WebSphere. All WebCenter Portal applications require an Oracle Internet Directory LDAP server.

Table 5–1 (Cont.) Getting the Spaces Application Up and Running on IBM WebSphere - Simple Topology

Task and link to more information	Mandatory or Optional?	Notes
Configure optional security components: ■ Configure discussions admin user ■ Configure activity graph engine admin user ■ Configure pagelet producer admin user ■ Configure user registry settings ■ Configure the trust service for REST ■ Configure SSL ■ Configure single sign-in (SS0)	Optional	
Extend cell for Oracle WebCenter Content Server: ■ Extend cell for WebCenter Content (includes Content Server) ■ Configure Content Server for WebCenter Portal ■ Connect Spaces application to Content Server	Mandatory	Mandatory for content presenter, wikis and blogs, and recommended for the Documents service.
Install and configure back-end components for WebCenter Portal services: ■ Connect to analytics collector ■ Connect to BPEL server ■ Connect to discussion server ■ Connect to events server ■ Configure personalization ■ Connect to presence server ■ Connect to mail server ■ Configure SES Search ■ Configure Spaces workflows ■ Register portlet producers ■ Register pagelet producer	Optional	Mandatory for the WebCenter Portal services you want to use

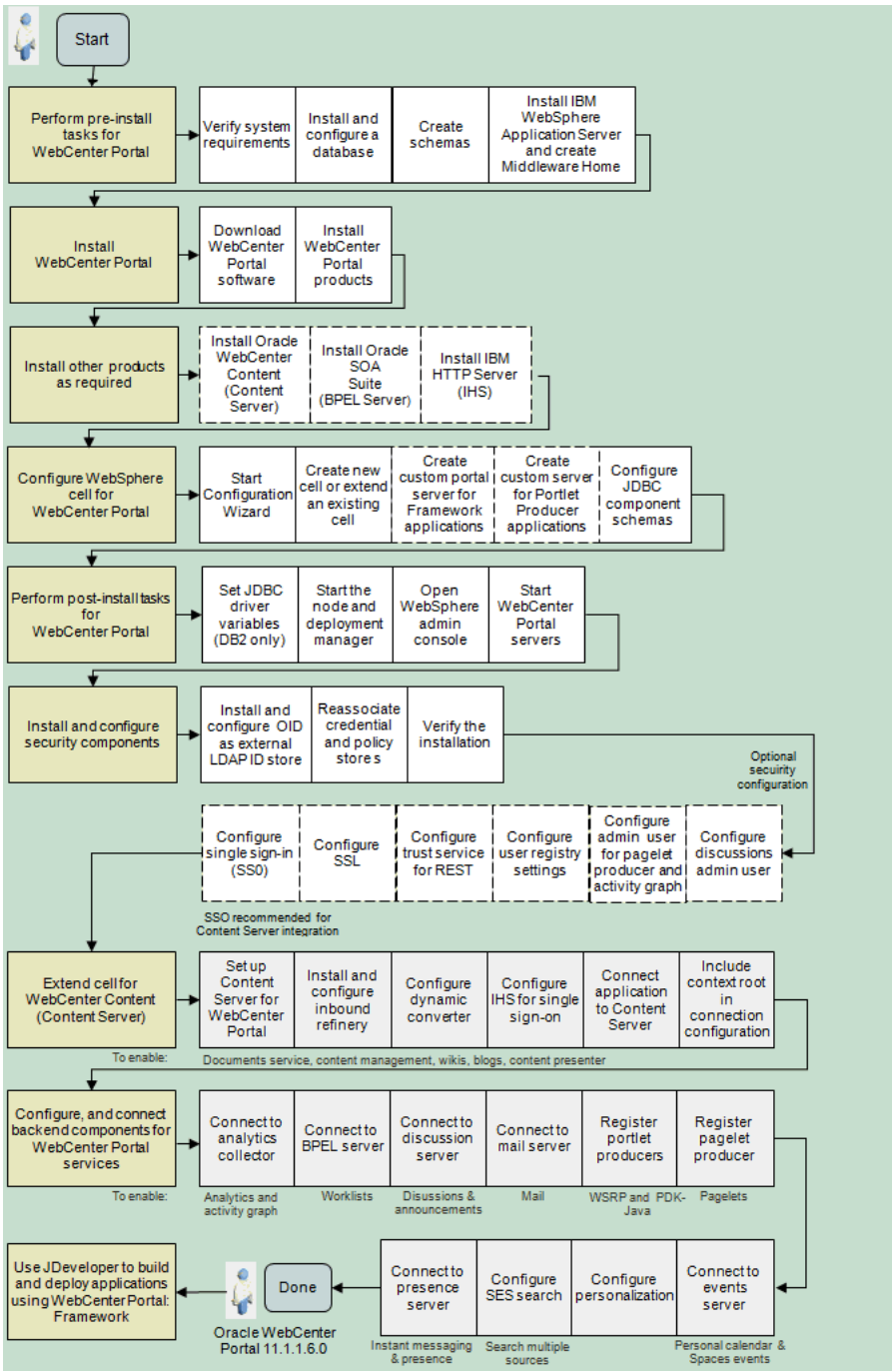
5.1.2 Creating a WebSphere Cell for Framework Application Deployments

Figure 5–2 illustrates the installation and configuration process if you want to build your own WebCenter Portal applications (referred to as Framework application) and deploy them in a simple, non-clustered environment.

Note: For deployment in a clustered environment, see Section 3.4, "Configuring Oracle Fusion Middleware High Availability on IBM WebSphere" and Section 5.2.19, "Configuring WebCenter Portal Applications for High Availability on IBM WebSphere".

Click the flow chart or use Table 5–2 to navigate to the appropriate documentation.

Figure 5–2 Creating a WebSphere Cell for Framework Application Deployments



Note: For deployment in a clustered environment, see Section 3.4, "Configuring Oracle Fusion Middleware High Availability on IBM WebSphere" and Section 5.2.19, "Configuring WebCenter Portal Applications for High Availability on IBM WebSphere".

Click the flow chart or use Table 5–2 to navigate to the appropriate documentation.

Table 5–2 Creating a WebSphere Cell for Framework Application Deployments - Simple Topology

Task and link to more information	Mandatory or Optional?	Notes
Verify system requirements	Mandatory	
Install and configure a database	Mandatory	
Create schemas for Framework applications	Mandatory	
Install IBM WebSphere Application Server and create Middleware Home	Mandatory	
Install Oracle WebCenter Portal	Mandatory	
Install other products as required: ■ Oracle WebCenter Content ■ Oracle SOA Suite ■ IBM HTTP Server (IHS)	Optional	Oracle WebCenter Content is mandatory for content presenter, wikis and blogs, and the Documents service. SOA is mandatory for the Worklist service and Spaces workflows. IHS is recommended for Oracle WebCenter Content Server integration and for single sign-on (SSO) since SSO is needed to stop multiple login prompts), and is required for REST and SOA.
Create new WebSphere cell for WebCenter Portal: ■ Create a custom managed server for Framework applications ■ Create a custom managed server for Portlet Producer applications	Mandatory	Oracle does not recommend deploying WebCenter Portal applications or WebCenter Portal Producer applications to the Administration Server or any of the default managed servers created during Oracle WebCenter Portal installation.
Perform post-install tasks for WebCenter Portal: ■ Set JDBC driver variables (DB2 only) ■ Start the node and deployment manager ■ Open WebSphere Admin console ■ Start WebCenter Portal servers	Mandatory	
Install and configure mandatory security components: ■ Install and configure Oracle Internet Directory (OID) as an external LDAP ID store ■ Reassociate credential and policy stores ■ Verify the WebCenter Portal installation	Mandatory	An external LDAP server is mandatory on IBM WebSphere. All WebCenter Portal applications require an Oracle Internet Directory LDAP server.

Table 5–2 (Cont.) Creating a WebSphere Cell for Framework Application Deployments - Simple Topology

Task and link to more information	Mandatory or Optional?	Notes
Configure optional security components: ■ Configure discussions admin user ■ Configure activity graph engine admin user ■ Configure pagelet producer admin user ■ Configure user registry settings ■ Configure the trust service for REST ■ Configure SSL ■ Configure single sign-in (SS0)	Optional	
Extend cell for Oracle WebCenter Content Server: ■ Extend cell for Oracle WebCenter Content (includes Content Server) ■ Configure Content Server for WebCenter Portal ■ Connect WebCenter Portal application to Content Server	Optional	Mandatory for content presenter, wikis and blogs, and the Documents service.
Configure, and connect back-end components for WebCenter Portal: ■ Connect to analytics collector ■ Connect to BPEL server ■ Connect to discussion server ■ Connect to events server ■ Connect to mail server ■ Connect to presence server ■ Configure personalization ■ Configure SES search ■ Register portlet producers ■ Register pagelet producer		
Use JDeveloper to build and deploy applications using WebCenter Portal: Framework		

5.2 Differences Installing and Configuring WebCenter Portal on IBM WebSphere

This section describe differences between installing and configuring WebCenter Portal install on WebLogic Server and IBM WebSphere:

- Installing WebCenter Portal Products on IBM WebSphere
- Configuring an IBM WebSphere Cell for the Spaces Application
- Configuring an IBM WebSphere Cell for Framework Applications
- Configuring an IBM WebSphere Cell for Portlet Producer Applications
- Performing General Post-install Tasks for WebCenter Portal on WebSphere

- Installing and configuring mandatory security components:
  - Installing External LDAP ID Store for WebCenter Portal Applications
  - Configuring an Admin User for the Spaces Application
  - Configuring an Admin User for the Discussions Server
  - Configuring an Admin User for Pagelet Producer and Activity Graph Applications
  - Reassociating the Credential and Policy Store
  - Setting Cookie Paths for WebCenter Portal Application Modules Post Deployment
  - Verifying the WebCenter Portal Installation on IBM WebSphere
- Optional security configuration:
  - Configuring User Registry Settings for External LDAP ID Store
  - Configuring Trust Service Information for the REST Service
  - Installing and Configuring IBM HTTP Server
  - Configuring Single Sign-On for WebCenter Portal Applications
  - Configuring SSL for WebCenter Portal Applications
- Cloning WebCenter Portal Installations on IBM WebSphere
- Configuring WebCenter Portal Applications for High Availability on IBM WebSphere

5.2.1 Installing WebCenter Portal Products on IBM WebSphere

Use the WebCenter Portal installer to install the binaries for *all* WebCenter Portal products on IBM WebSphere. The instructions are similar to those provided for Oracle WebLogic Server in "Installing Oracle WebCenter Portal" in *Oracle Fusion Middleware Installation Guide for Oracle WebCenter Portal*.

There are a few differences when installing on IBM WebSphere. For details see Section 2.5.2, "Special Instructions When Installing Oracle Fusion Middleware with IBM WebSphere".

5.2.2 Configuring an IBM WebSphere Cell for the Spaces Application

To configure an IBM WebSphere cell for the Spaces application:

1. Start the IBM WebSphere version of Oracle Fusion Middleware Configuration Wizard:

   ```
   WC_ORACLE_HOME/common/bin/was_config.sh
   ```

   For details, see "Using the Configuration Wizard" in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

2. On the **Select Domain Source** screen, select **Oracle WebCenter Spaces** and any other WebCenter Portal products you want to install, such as the Discussions Server, Portlet Producers, Analytics Collector, and so on.

   For details, see "Selecting Oracle WebCenter Portal Products for Configuration" in the *Oracle Fusion Middleware Installation Guide for Oracle WebCenter Portal*.

5.2.3 Configuring an IBM WebSphere Cell for Framework Applications

If you want to deploy applications built using WebCenter Portal: Framework to an IBM WebSphere application server you must configure a suitable server using WebCenter Portal's *Custom Portal Template*.

For Custom Portal Template to display in the Oracle Fusion Middleware Configuration Wizard you need to set a system property before you run the Configuration Wizard:

> **Note:** This step is not required if you are configuring a cell for the Spaces application.

1. Set the **JVM_ARG** environment variable:

   ```
   setenv CONFIG_JVM_ARGS -DTemplateCatalog.enable.selectable.all=true
   ```

2. Start the IBM WebSphere version of Oracle Fusion Middleware Configuration Wizard using: `WC_ORACLE_HOME/common/bin/was_config.sh`.

   For details, see "Using the Configuration Wizard" in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

3. On the **Select Domain Source** screen, select **Base this domain on an existing template**, and click **Browse** to locate the template:

   - On UNIX operating systems, the template is located at:

     WC_ORACLE_HOME`/common/templates/was/oracle.wc_custom_portal_template_11.1.1.jar`

   - On Windows operating systems, the template is available here:

     WC_ORACLE_HOME`\common\templates\was\oracle.wc_custom_portal_template_11.1.1.jar`

   For details, see "Creating a Custom Managed Server for Framework Applications" in the *Oracle Fusion Middleware Installation Guide for Oracle WebCenter Portal*.

5.2.4 Configuring an IBM WebSphere Cell for Portlet Producer Applications

If you want to deploy Portlet Producer applications built using WebCenter Portal: Framework to an IBM WebSphere application server you must configure a suitable server using WebCenter Portal's *Custom Services Producer Template*.

For Custom Services Producer Template to display in the Oracle Fusion Middleware Configuration Wizard you need to set a system property before you run the Configuration Wizard:

> **Note:** This step is not required if you are configuring a cell for the Spaces application.

1. Set the **JVM_ARG** environment variable:

   ```
   setenv CONFIG_JVM_ARGS -DTemplateCatalog.enable.selectable.all=true
   ```

2. Start the IBM WebSphere version of Oracle Fusion Middleware Configuration Wizard:

WC_ORACLE_HOME/common/bin/was_config.sh

For details, see "Using the Configuration Wizard" in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

3. On the **Select Domain Source** screen, select **Base this domain on an existing template**, and click **Browse** to locate the template:
   - On UNIX operating systems, the template is located at:

     WC_ORACLE_HOME/common/templates/was/oracle.wc_custom_services_producer_template_11.1.1.jar
   - On Windows operating systems, the template is available here:

     WC_ORACLE_HOME\common\templates\was\oracle.wc_custom_services_producer_template_11.1.1.jar

   For details, see "Creating a Custom Managed Server for Portlet Producer Applications" in the *Oracle Fusion Middleware Installation Guide for Oracle WebCenter Portal*.

5.2.5 Performing General Post-install Tasks for WebCenter Portal on WebSphere

This section includes the following subsections:

- Setting JDBC Driver Variables (DB2 only)
- Starting the Node Agent and Deployment Manager
- Opening IBM WebSphere Administrative Console
- Starting WebCenter Portal Servers

5.2.5.1 Setting JDBC Driver Variables (DB2 only)

If you are using a DB2 database, you must set the following environment variables to include the full path to `db2jcc4.jar`, `db2jcc_license_cu.jar` and `db2jcc_license_cisuz.jar`:

- DB2_JCC_DRIVER_NATIVEPATH
- DB2_JCC_DRIVER_PATH

You must do this immediately after installing WebCenter Portal products using the Configuration Wizard. If you do not do this, all DB2 connection tests will fail.

If you are deploying your own WebCenter Portal applications to IBM WebSphere, you must also set these two environment variables at the Deployment Manager scope. If you do not, the JDeveloper MDS deployment wizard cannot query or allow configuration to DB2 back-end MDS repositories, and this causes issues at application runtime.

To set DB2 driver environment variables:

1. Log in to the IBM WebSphere Administrative Console:

   https://*host*:*port*/ibm/console

2. Navigate to **Environment > WebSphere variables**
3. Set DB2 driver variables for the server node:
   a. From the **Scope** drop down, select the node containing your WebCenter Portal installation.

b. Locate and set the following JBDC variables:

`DB2_JCC_DRIVER_NATIVEPATH`

`DB2_JCC_DRIVER_PATH`

Specify the location of the required DB2 drivers (`db2jcc4.jar`, `db2jcc_license_cu.jar` and `db2jcc_license_cisuz.jar`).

Refer to your IBM WebSphere documentation to find the location of these drivers. Look for the topic entitled "*Data source minimum required settings for DB2 with the application server*" or similar.

c. Save both settings.

4. If you are using a cluster, repeat step 3 for each node in the cluster.
5. To test the DB2 connection:
   a. Navigate to **Resources** > **JDBC** >**Data sources**.
   b. Select a data source in the table, and click **Test Conection**.
6. If you are deploying your own WebCenter Portal applications to IBM WebSphere, repeat step 3 at the Deployment Manager scope.
   a. From the **Scope** drop down, select the **Node=ManagerNode, Server=dmgr** scope where ManagedNode maps to the Manage Node of your installation.
   b. Create and set JBDC variables, as above:

      `DB2_JCC_DRIVER_NATIVEPATH`

      `DB2_JCC_DRIVER_PATH`

   c. Save both settings.

5.2.5.2 Starting the Node Agent and Deployment Manager

After using the Configuration Wizard to install and configure Oracle WebCenter Portal products on IBM WebSphere, start up the deployment manager for the cell and the application node as described in Section 2.8, "Task 8: Start the IBM WebSphere Servers".

The IBM WebSphere Administration Console is accessible after starting the node and deployment manager.

5.2.5.3 Opening IBM WebSphere Administrative Console

IBM WebSphere Administrative Console provides a Web-based interface for managing the IBM WebSphere environment. The IBM WebSphere Administrative Console is similar to Oracle WebLogic Server Administration Console, that is, while you cannot use the console to manage *Oracle WebCenter Portal products*, you can use the console to monitor and manage the cell and the servers on which Oracle WebCenter Portal and other Oracle Fusion Middleware products are deployed. For more information, see Section 3.1.1, "Using the WebSphere Administrative Console".

5.2.5.4 Starting WebCenter Portal Servers

After installing and configuring Oracle WebCenter Portal on IBM WebSphere and starting both the deployment manager and node, you can start the WebCenter Portal servers using the IBM WebSphere Administrative Console or Fusion Middleware Control. For details, see Section 2.8, "Task 8: Start the IBM WebSphere Servers".

The default names for WebCenter Portal servers in a WebCenter Portal:Spaces installation are:

- WC_Spaces
- WC_Collaboration
- WC_Portlet
- WC_Utilities

The default names for WebCenter Portal servers for Framework applications and portlet producer deployments are:

- WC_CustomPortal (Framework applications)
- WC_CustomServicesProducer (Portlet producer applications)

5.2.6 Installing External LDAP ID Store for WebCenter Portal Applications

An LDAP server is not automatically installed and configured when you install Oracle WebCenter Portal products on IBM WebSphere. Before you can configure WebCenter Portal, you must install and configure Oracle Internet Directory (OID) as the external LDAP ID store for your WebCenter Portal applications. For instructions on how to set up external LDAP ID stores, such as Oracle Internet Directory, see Section 8.1, "IBM WebSphere Identity Stores".

Note: All WebCenter Portal applications, including Spaces, must use Oracle Internet Directory (OID).

Once the LDAP ID store is set up, you must set the CONNECTION_POOL_CLASS property in cell's `jps-config.xml`. For details, Section 5.2.6.1, "Setting the Connection Pool on IBM WebSphere When Connecting to an External LDAP Server".

5.2.6.1 Setting the Connection Pool on IBM WebSphere When Connecting to an External LDAP Server

To avoid excessive database connections, you must add the following `<serviceInstance>` entry to the cell's `jps-config.xml`:

```
<property name="CONNECTION_POOL_CLASS"
value="oracle.security.idm.providers.stdldap.JNDIPool"/>
```

1. Modify `jps-config.xml` using a text editor:
   a. Open the following file:

      WAS_HOME/profiles/*dmgr_profile_name*/
      config/cells/myCell/fmwconfig/jps-config.xml

      Where `Dmgr01` maps to the Deployment Manager name, and `myCell` maps to the cell name.

   b. Specify the following:

```
<serviceInstance name="idstore.ldap.0" provider="idstore.ldap.provider">
            <property name="subscriber.name"
value="dc=us,dc=oracle,dc=com"/>
            <property name="CONNECTION_POOL_CLASS"
value="oracle.security.idm.providers.stdldap.JNDIPool"/>
            <property name="bootstrap.security.principal.key"
value="bootstrap_idstore"/>
```

```
            <property name="idstore.type" value="OID"/>
            <property name="ldap.url" value="ldap://example.com:3060"/>
            <property name="bootstrap.security.principal.map"
value="BOOTSTRAP_JPS"/>
            <property name="user.login.attr" value="mail"/>
            <property name="username.attr"  value="mail"/>
            <extendedProperty>
                <name>user.search.bases</name>
                <values>
                    <value>cn=Users,dc=us,dc=oracle,dc=com</value>
                </values>
            </extendedProperty>
            <extendedProperty>
                <name>group.search.bases</name>
                <values>
                    <value>cn=Groups,dc=us,dc=oracle,dc=com</value>
                </values>
            </extendedProperty>
        </serviceInstance>
```

2. Restart all the servers.

5.2.7 Configuring an Admin User for the Spaces Application

After installing Oracle WebCenter Portal products on IBM WebSphere and setting up your LDAP ID store, you must manually grant the Spaces administrator role to a user in the ID store.

You can configure the administrative user through Fusion Middleware Control or use the `Opss.grantAppRole` wsadmin command as shown in this example:

```
Opss.grantAppRole(appStripe='webcenter', appRoleName='s8bba98ff_
4cbb_40b8_beee_
296c916a23ed#-#Administrator',principalClass='weblogic.security.
principal.WLSUserImpl', principalName='myadmin')
```

For more information, see "Granting the Spaces Administrator Role" in *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal*.

> **Note:** *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal* describes how to run the equivalent WebLogic WLST command. The way you run IBM WebSphere wsadmin commands is slightly different to WLST, so if you are new to wsadmin, refer to Section 5.4.1, "Running WebCenter Portal wsadmin Commands".

5.2.8 Configuring an Admin User for the Discussions Server

If you chose to install **Oracle WebCenter Portal's Discussion Server** while installing Oracle WebCenter Portal on WebSphere you must configure an administrative user for the discussions server using the wsadmin command `WebCenter.addDiscussionsServerAdmin`.

For example:

```
WebCenter.addDiscussionsServerAdmin(appName='owc_discussions_
11.1.1.4.0',name='myadmin',type='USER')
```

Where:

- `myadmin` is a user in the identity store with administrative privileges in the WebCenter Portal application.
- `owc_discussions_11.1.1.4.0` is the name of the discussion server application installed on IBM WebSphere.

For information on how to run WebCenter Portal wsadmin commands, see Section 5.4.1, "Running WebCenter Portal wsadmin Commands".

See also, "addDiscussionsServerAdmin" in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

Note: After adding an admin user using wsadmin you must restart the WC_Collaboration server.

5.2.9 Configuring an Admin User for Pagelet Producer and Activity Graph Applications

If you chose to install **Oracle WebCenter Pagelet Producer** or **Oracle WebCenter Activity Graph Engines** while installing Oracle WebCenter Portal on WebSphere you must assign administrative permissions to an appropriate user or a group through the following roles:

Application Name	Admin Role
pageletproducer	EnsembleAdmin
activitygraph-engines	activity-graph-admins

To configure administrators:

1. Log in to the IBM WebSphere Administrative Console.
2. Navigate to **Applications > Application Types > WebSphere enterprise applications**.
3. Configure an administrative user for the Pagelet Producer admin application:
   a. Select **pageletproducer**.
   b. Click **Security role to user/group mapping**.
   c. Select the **EnsembleAdmin** role and the click either **Map Users...** or **Map Groups...** to assign one or more users/groups to this admin role.
   d. Click **OK**.
4. Configure administrative user for the Activity Graph Engine application:
   a. Select **activitygraph-engines_11.1.1.6.0**.
   b. Click **Security role to user/group mapping**.
   c. Select the **activity-graph-admins** role and the click either **Map Users...** or **Map Groups...** to assign one or more users/groups to this admin role.
   d. Click **OK**.
5. Restart **WC_Portlet** (pageletproducer) and **WC_Utilities** (activitygraph-engines), as required.

5.2.10 Reassociating the Credential and Policy Store

When you install WebCenter Portal products on IBM WebSphere Application Server - Network Deployment (ND), you must reassociate your policy store with an external LDAP (either Oracle Internet Directory 11gR1 or 10.1.4.3), or a database. Note that when using an external LDAP-based store, the credential store and policy store must be configured to use the same LDAP server. For detailed steps see "Configuring the Policy and Credential Store" in *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal*.

5.2.11 Setting Cookie Paths for WebCenter Portal Application Modules Post Deployment

By default, applications deployed on IBM WebSphere have their cookie path set to "/". This default setting means that all applications on the same IBM WebSphere cell share the same session identifier and therefore, as you move between applications, the session identifier value for the previous application is overwritten. For example, if you access the Spaces application (`/webcenter`), access Enterprise Manager (`/em`), and then move back to Spaces (`/webcenter`) you are prompted to log in to Spaces again because the previous session identifier value is overwritten at the point when you log in to Enterprise Manager (`/em`),

To avoid session invalidation as you move between applications, specify a unique cookie path for each application following the steps below.

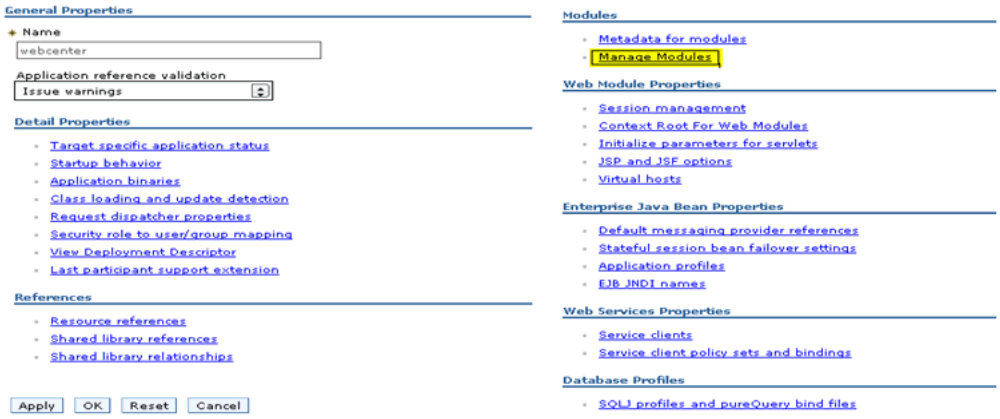
1. Log in to the IBM WebSphere Administrative Console.

   `https://host:port/ibm/console`

2. Navigate to **Applications > WebSphere enterprise applications**.
3. Select the name of your application from the list.

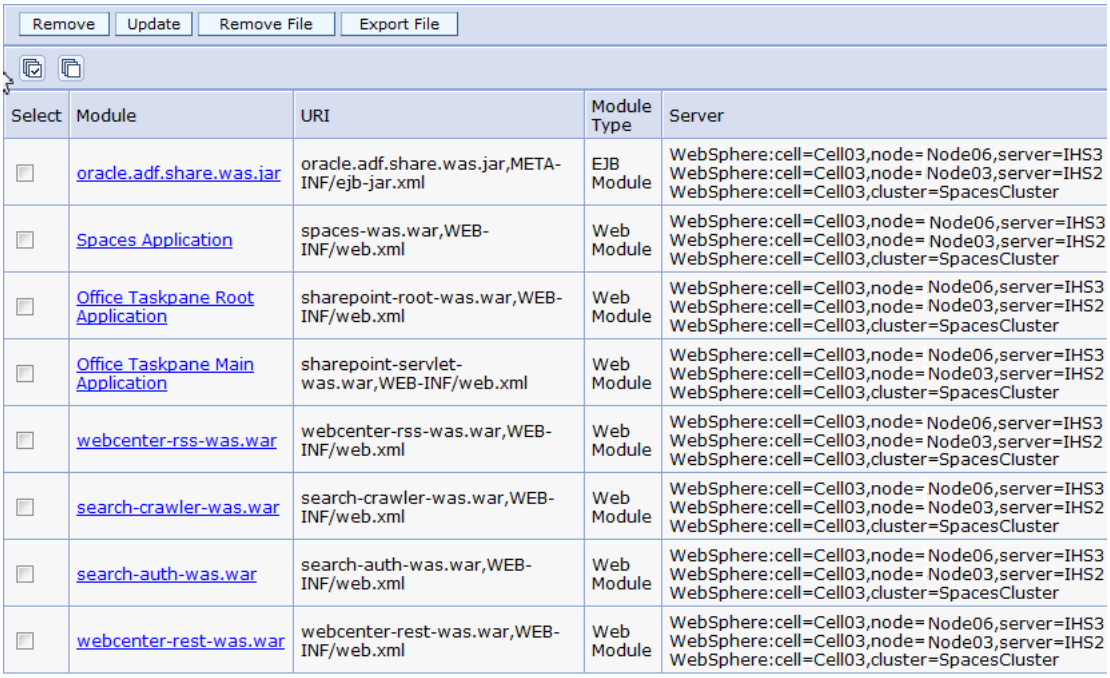
   For example, the name of the Spaces application is `webcenter`.
4. Click **Manage Modules** (Figure 5–3).

Figure 5–3 Enterprise Applications - Manage Modules



A list of modules displays (Figure 5–4).

Figure 5–4 List of Modules to Manage



5. Set the cookie path for each module listed in Table 5–3:

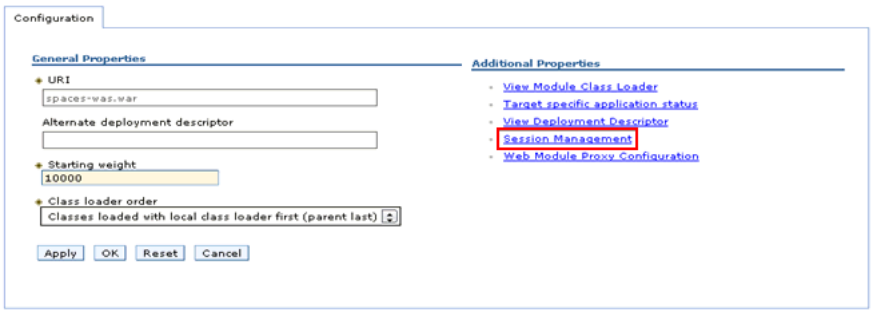
Table 5–3 Cookie Paths for WebCenter Portal Modules

Module Name	Cookie Path
spaces-was.war	/webcenter
webcenter-rest-was.war	/rest
search-crawler-was.war	/rsscrawl
webcenter-rss-was.war	/rss
sharepoint-servlet-was.war	/wcsdocs

   a. Click the module name, for example **spaces-was.war** (Spaces Application).

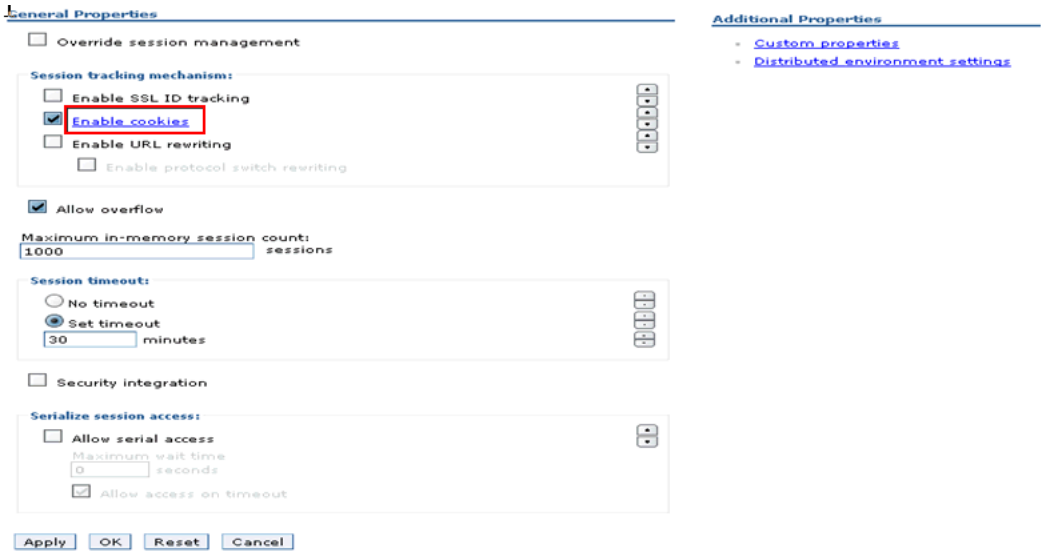
   b. Click **Session Management** (Figure 5–5).

Figure 5–5 Configure Module



   c. Select the **Enable cookies** check box (Figure 5–6).

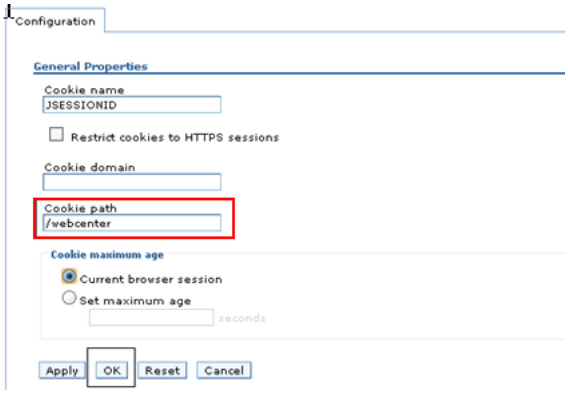
Figure 5–6 Configure Module - Enable Cookies



d. Click **Enable cookies** link.

e. Enter the appropriate **cookie path** for the selected module (Figure 5–7). For details, see Table 5–3.

Figure 5–7 Configure Module - Set Cookie Path



f. Click **OK** and then **Save**.

g. Select the **Override session management** check box, and then click **OK**.

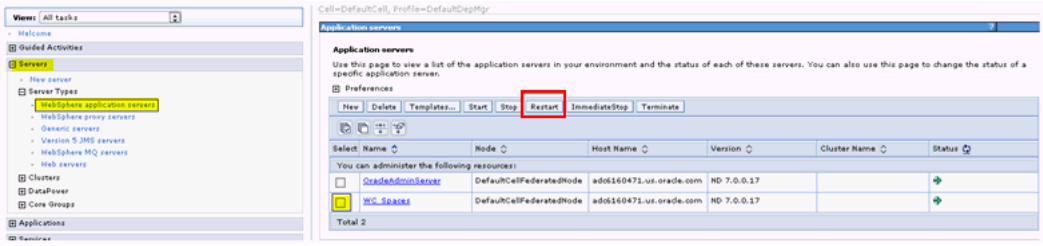
h. Repeat steps a through f for each module listed in Table 5–3.

6. Restart the server on which the Spaces application is deployed.

a. Navigate to **Servers** > **WebSphere Application servers.**

b. Select the **WC_Spaces** check box, and click **Restart** (Figure 5–8).

Figure 5–8 Restart WC_Spaces Server



5.2.12 Verifying the WebCenter Portal Installation on IBM WebSphere

To verify your WebCenter Portal installation, start your browser and enter the following URLs:

- To access the IBM WebSphere Administration Server console:

  `https://dmgr_server_host:WC_Adminhost_port/ibm/console`

  You will be prompted for the username and password credentials that you specified on the *Specify Deployment Manager Information* screen of the Configuration Wizard.

  To discover the port numbers required to access individual servers, such as WC_Spaces, OracleAdminServer, and so on, navigate to the server in the IBM WebSphere Administration Server console, select the **Ports** link and look for "**WC_defaulthost**". See also, Section 3.1.1.2, "Locating the Port Number and URL of the IBM WebSphere Administrative Console".

- To access Enterprise Manager:

  `http://OracleAdminServer_host:OracleAdminServer_port/em`

- To access WebCenter Portal's Spaces application:

  `http://WC_Spaces_server_host:WC_Spaces_server_port/webcenter`

  The default port number for the Spaces application is 8888.

- To access Pagelet Producer:

  `http://WC_Portlet_server_host:WC_Portlet_server_port`

  The default port number for Pagelet Producer is 8889.

  To access the Pagelet Producer console:

  `http://WC_Portlet_server_host:WC_Portlet_server_port/pageletadmin`

- To access Oracle WebCenter Analytics Collector, Oracle WebCenter Activity Graph Engines and Oracle WebCenter Personalization:

  `http://WC_Utilities_server_host:WC_Utilities_server_port/activitygraph-engines`

  To access Oracle WebCenter Activity Graph Engines:

  `http://WC_Utilities_server_host:Wc_Utilities_server_port/activitygraph-engines/Login.jsp`

  To access Oracle WebCenter Personalization:

  `http://WC_Utilities_server_host:Wc_Utilities_server_port/wcps/api/property/resourceIndex`

The default port number for Oracle WebCenter Analytics Collector, Oracle WebCenter Activity Graph Engines and Oracle WebCenter Personalization is 8891.

- To access WebCenter OmniPortlet and Web Clipping Portlets:

  `http://`*`WC_Portlet_server_host`*`:`*`WC_Portlet_server_port`*`/portalTools/`

  The default port number for WebCenter Portlets is 8889.

- To access Oracle WebCenter Portal's Discussion Server:

  `http://`*`WC_Collaboration_server_host`*`:`*`WC_Collaboration_server_port`*`/owc_discussions`

  The default port number for the discussions server is 8890.

5.2.13 Configuring User Registry Settings for External LDAP ID Store

Several additional user registry settings may be required after configuring the external LDAP ID store:

- **Enable nested user group searching** - IBM WebSphere supports nested user groups however, they are not automatically included in LDAP searches as enabling them impacts performance. If your WebCenter Portal installation utilizes nested user groups you can enable this feature.
- **Configure the user login attribute** - If required, you can set the user login attribute to an attribute other than `cn`. For example, if set to `mail`, LDAP searches utilize the username that is used to log in to the WebCenter Portal application.

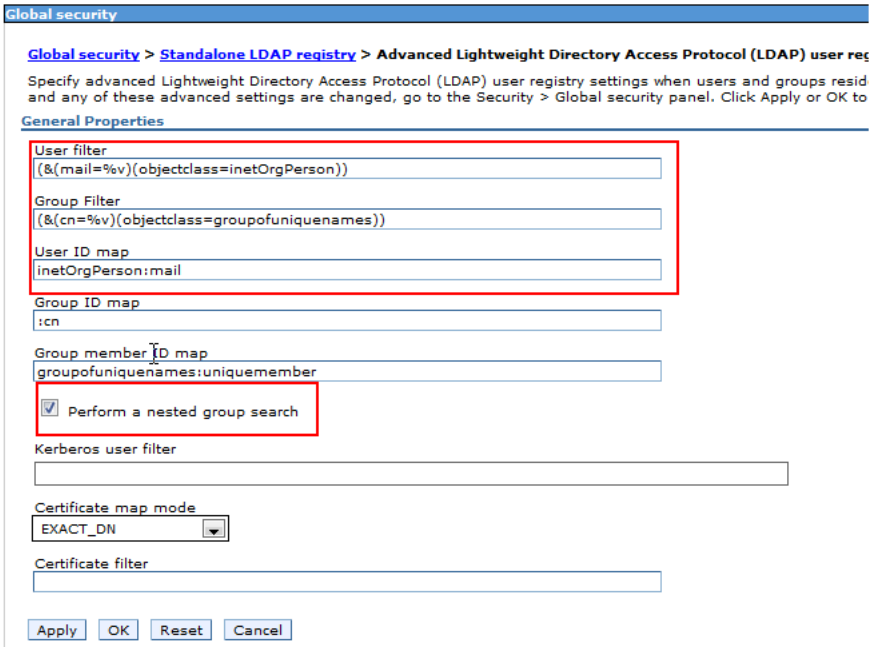
To configure these settings:

1. Log in to the IBM WebSphere Administrative Console.
2. Navigate to **Global security > Standalone LDAP registry**.
3. Under Additional properties, click **Advanced Lightweight Directory Access Protocol (LDAP) user registry settings**.
4. Specify the user login attribute in **User filter** and **User ID map** (Figure 5–9).

   For example, to configure the `mail` attribute, enter:

   - **User filter - (&(mail=%v)(objectclass=inetOrgPerson))**
   - **User ID map - inetOrgPerson:mail**

Figure 5–9 Advanced LDAP User Registry Settings



5. Select the **Perform a nested group search** check box.
6. Click **OK**.
7. Modify `jps-config.xml` using a text editor:
   a. Open the `MW_HOME/user_projects/domains/my_domain/config/fmwconfig/jps-config.xml`
   b. Specify the user login attribute in the LDAP properties **user.login.attr** and **username.attr** to **mail**.

      For example, to configure the `mail` attribute, enter:

```
<serviceInstance provider="idstore.ldap.provider" name="idstore.ldap.0">
          <!-- existing props ... ->
          <property name="user.login.attr" value="mail"/>
          <property name="username.attr"  value="mail"/>
          <extendedProperty>
              ... ...
          </extendedProperty>
      </serviceInstance>
```

   c. Restart all the servers.

5.2.14 Configuring Trust Service Information for the REST Service

This section describes how to configure an identity asserter for the REST service.

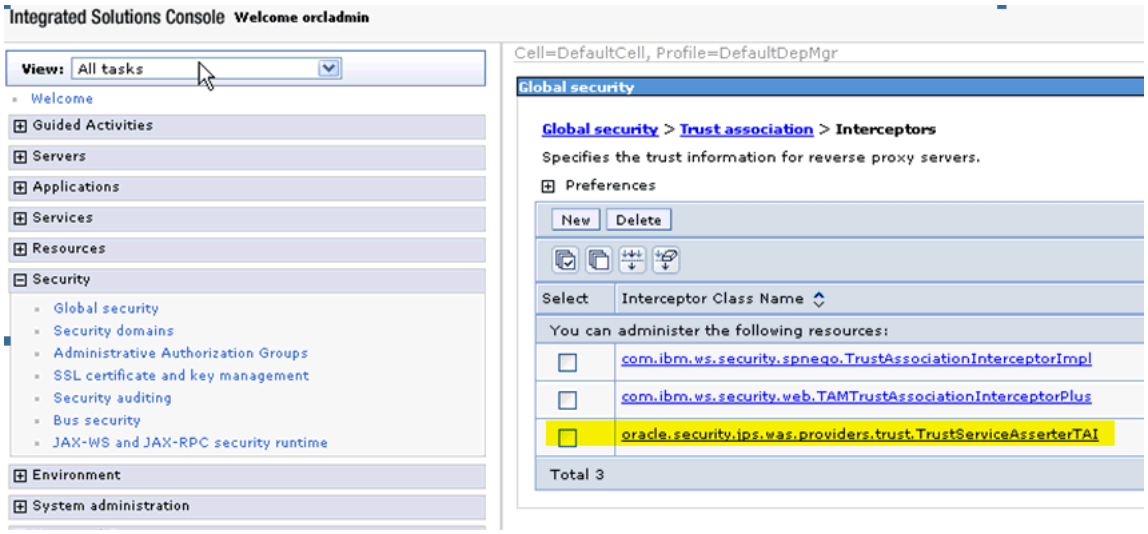
1. Login to the IBM WebSphere Administrative Console.
2. Navigate to **Security > Global Security.**
3. In the **Authentication** section, expand **Web and SIP** security, and then click **Trust Association**.
4. Select the **Enable trust association** check box and save the changes.
5. In the **Additional Properties** section, click **Interceptors**.

6. Click **New.**
7. For **Interceptor class name**, enter the fully qualified Trust Service TAI name:

   `oracle.security.jps.was.providers.trust.TrustServiceAsserterTAI`

   The Trust Association Interceptor (TAI) class is in the `jps-was.jar` file located in the standard Oracle Platform Security Services (OPSS) jar distribution directory.
8. Save the changes (Figure 5–10).

Figure 5–10 Configuring Trust Information



5.2.15 Installing and Configuring IBM HTTP Server

This section describes how to install and configure an IBM HTTP Server to front end the WebSphere Application Server hosting Oracle WebCenter Portal. An IBM HTTP server is required to implement single sign-on for WebCenter Portal applications (Spaces and Framework applications) and also for high availability environments. For more information about using the IBM HTTP Server, see:

- Section 5.2.16, "Configuring Single Sign-On for WebCenter Portal Applications"
- Section 5.2.19, "Configuring WebCenter Portal Applications for High Availability on IBM WebSphere"

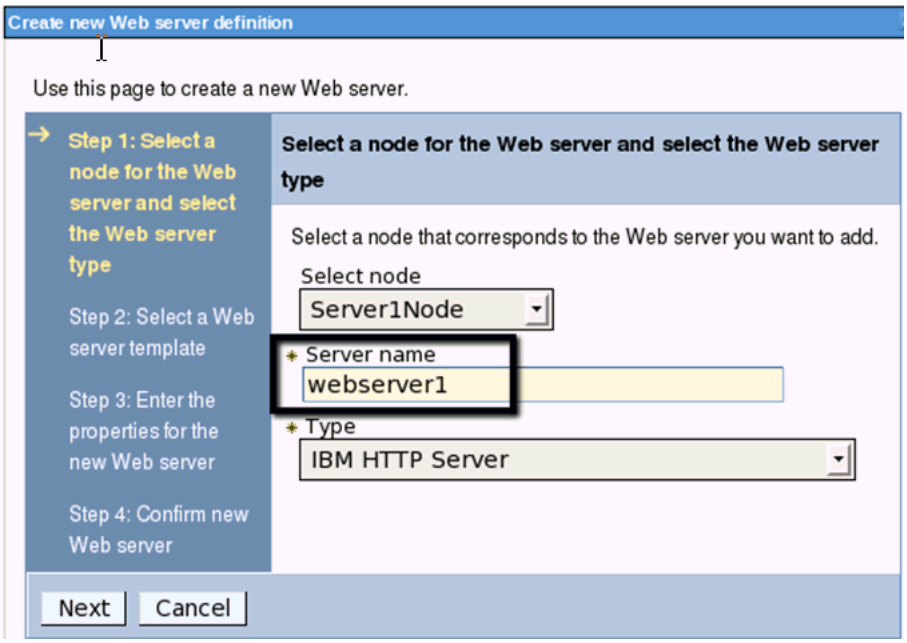
To install and configure an IBM HTTP Server:

1. Install the IBM HTTP Server, and take note of the **Server Name** and the **HTTP Port**.

   For detailed installation instruction, refer to IBM HTTP Server documentation. See also, Section 1.4, "Documentation Resources for Using Oracle Fusion Middleware on IBM WebSphere".
2. Configure the HTTP Server, specifying the server name and port you specified in step 1.
   a. Log in to the IBM WebSphere Administrative Console.
   b. Navigate to **Servers >Web Servers > New**.
   c. Enter the **Server name** that you defined during IBM HTTP server installation. For example, **webserver1** (Figure 5–11).

Ensure that the server type is IBM HTTP Server.

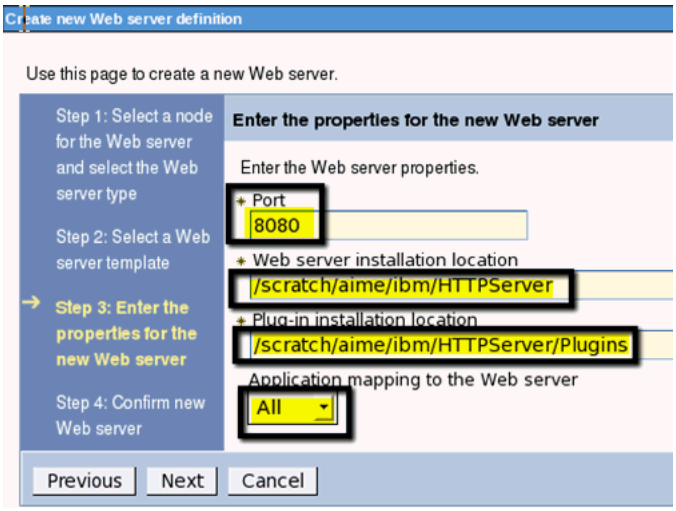
Figure 5–11 Configure HTTP Server - Server Name



d. Click **Next**

e. Enter the port that you defined during HTTP server installation. For example 8080 (Figure 5–12).

Figure 5–12 Configure HTTP Server - Port



f. Click **Next** and **Finish**.

3. Create a virtual host entry to enable access to the port specified in step 1.

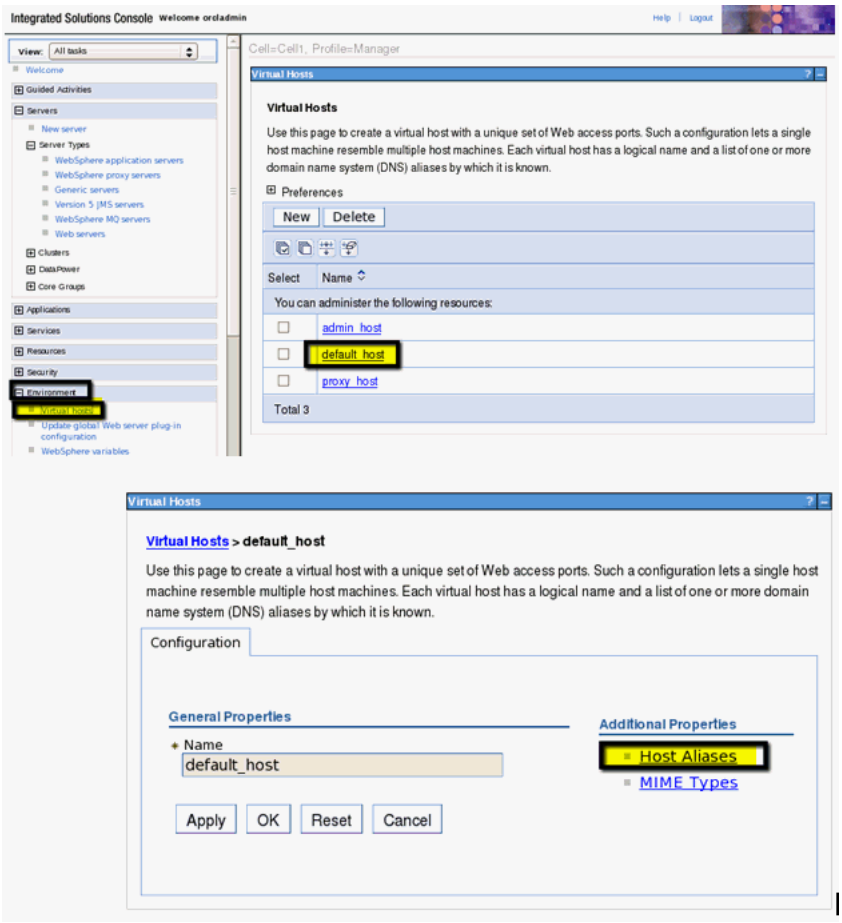
If the port is *not* accessible, error messages similar to that shown display:

```
SRVE0255E: A WebGroup/Virtual Host to handle /webcenter/ has
not been defined.
```

```
SRVE0255E: A WebGroup/Virtual Host to handle host:8080 has
not been defined.
```

a. In the WebSphere Administrative Console, navigate to **Environment, Virtual Hosts> default_host> Host Aliases** (Figure 5–13).

Figure 5–13 Configure Virtual Host -default_host

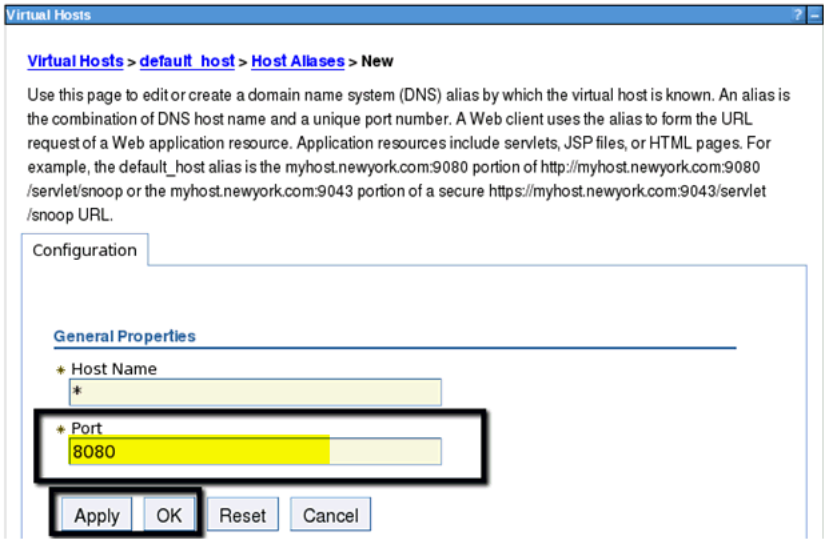


b. On the Host Aliases page, select **New**.

c. For **Port**, enter the port number specified in step 1 (Figure 5–14).

Leave **Host Name** as *.

Figure 5–14 Configure default_host - Port



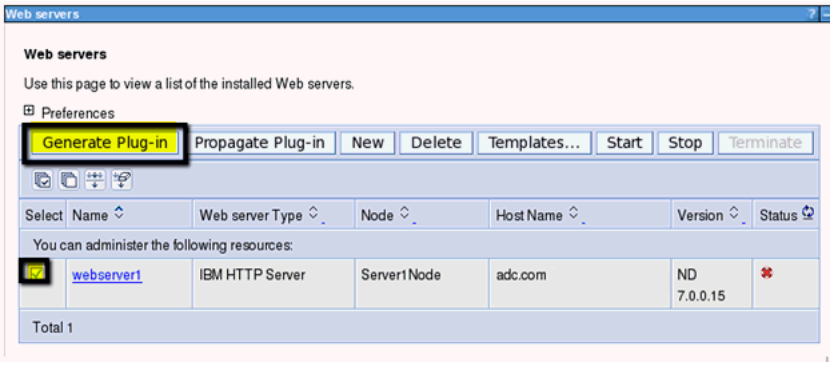
4. Generate and propagate the Web Server plug-in that coordinates the wiring between the WebSphere Application Server and the HTTP Server front end.

   Note: You must repeat this step each time there is a change to applications deployed on the servers.

   a. In WebSphere Administrative Console, navigate to **Servers >Web Servers** and select the HTTP server. you created.

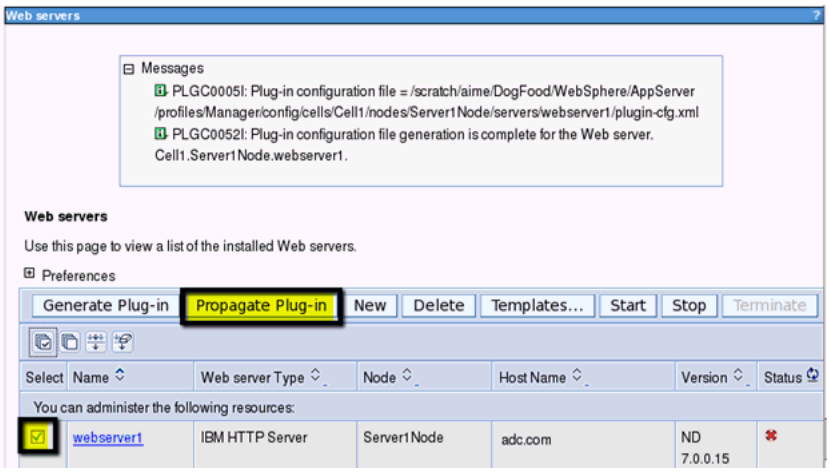
   b. Click **Generate Plug-in** (Figure 5–15).

Figure 5–15 Configure HTTP Server - Generate Plug-in



   c. Click **Propagate Plug-in** (Figure 5–16).

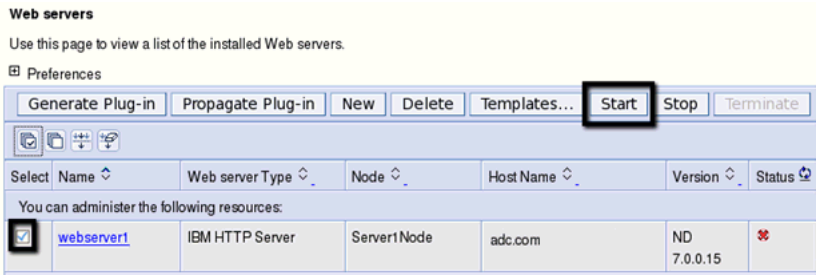
Figure 5–16 Configure HTTP Server - Propagate Plug-in



5. Update the Web Server `httpd.conf` file to refer to the correct plug-in file:
   a. Click the Web Server.
   b. Next to **Configuration file name**, click **Edit**.
   c. Scroll down to see the property **WebSpherePluginConfig** and confirm that it is pointing to the `plugin-cfg.xml` file under this Web Server's directory name (for example, `/IHS_HOME/Plugins/config/<WebServerName>/plugin-cfg.xml`).
   d. Click **Apply** and then **OK**.
   e. Click OK again to return to the Web Servers page.
6. Click **Start**.

   Note: If the instance is already running, click **Stop**, then **Start** (Figure 5–17).

Figure 5–17 Configure HTTP Server - Start



7. Restart the WebSphere server on which the WebCenter Portal application is deployed.
8. Restart the HTTP server to enable the virtual host and Web Server plug-in.

The WebCenter Portal application should now be accessible on the HTTP Server port (Figure 5–18):

Figure 5–18 HTTP Server Port - Application Accessible

5.2.16 Configuring Single Sign-On for WebCenter Portal Applications

This section includes the following sub sections:

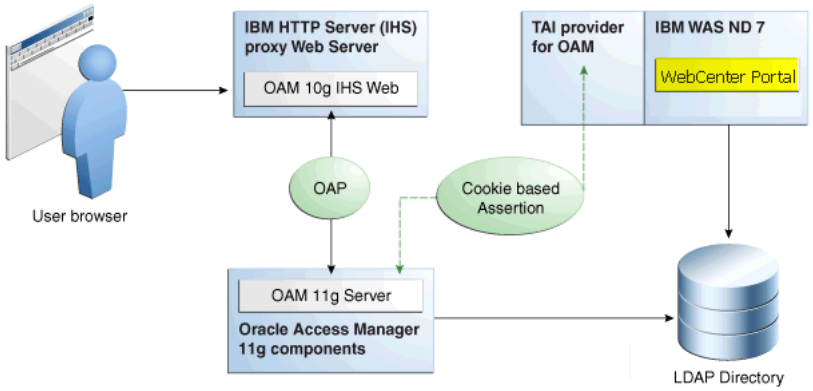
- Section 5.2.16.1, "Configuring OAM 11g Single Sign-On"
- Section 5.2.16.2, "Configuring WebCenter Portal Applications for Single Sign-On"

5.2.16.1 Configuring OAM 11g Single Sign-On

This section describes how to set up single sign-on for WebCenter Portal installations on IBM WebSphere, using Oracle Access Manager (OAM) 11g (Figure 5–19).

> **Note:** WebCenter Portal applications deployed on IBM WebSphere support Oracle Access Manager (OAM) 11g. Earlier versions, such as Oracle Access Manager (OAM) 10g, are not supported on IBM WebSphere.

Figure 5–19 Configuring OAM Single Sign-On for WebCenter Portal Installations on IBM WebSphere



Oracle Access Manager Identity Assertion Provider for IBM WebSphere can be used to provide authentication and single sign-on with Oracle Access Manager (OAM) 11g. Chapter 9, "Managing OAM Identity Assertion on IBM WebSphere" describes the Oracle Access Manager Identity Assertion Provider is detail. The purpose of this section is to guide you through single sign--on configuration requirements for WebCenter Portal applications. The main steps are:

- Install Oracle Access Manager 11g

- Install and configure IBM HTTP Server
- Register the WebGate agent
- Restart IHS
- Install WebGate 10g
- Configure IBM WebSphere for OAM single sign-on
- Configure logout details
- Configure WebCenter Portal applications to require certificate based authentication
- Configure other WebCenter Portal components for single sign-on

To set up single sign-on for WebCenter Portal applications, using OAM 11g:

1. **Install Oracle Access Manager 11g**.

   See Chapter 9, "Managing OAM Identity Assertion on IBM WebSphere".

2. **Install and configure IBM HTTP Server**.

   See Chapter 5.2.15, "Installing and Configuring IBM HTTP Server".

3. **Register the WebGate agent** on the machine where OAM 11g is installed *before* installing WebGate on IBM HTTP Server.

   You can register the WebGate agent using the OAM Console or, if you have administrator rights, you can use the `oamreg` tool. Follow the steps below to register the WebGate agent using the `oamreg` tool in inband mode.

   a. Navigate to the following directory on the Oracle Access Manager server:

      IDM_HOME`/oam/server/rreg/client/`

   b. On the command line, untar `RREG.tar.gz`

      ```
      gunzip RREG.tar.gz
      tar -xvf RREG.tar
      ```

      The tool used to register the agent is located in the following location:

      ```
      (UNIX) RREG_HOME/bin/oamreg.sh
      (Windows) RREG_HOME\bin\oamreg.bat
      ```

      Note: *RREG_HOME* is the directory where you extracted the contents of `RREG.tar.gz/rreg`.

   c. Set the following environment variables in the oamreg.sh or oamreg.bat script:

      OAM_REG_HOME - Set this variable to the absolute path to the directory where you extracted the contents of RREG.tar/rreg

      JDK_HOME - Set this variable to the absolute path to the directory where Java/JDK is installed on your machine.

   d. Change directories to *RREG_HOME*`/input` and copy the following files to this location:

      ```
      WC_ORACLE_HOME/webcenter/scripts/webcenter.oam.conf
      SOA_ORACLE_HOME/soa/prov/soa.oam.conf
      WC_CONTENT_ORACLE_HOME/common/security/oam.conf
      ```

e. Create a new file named `WebCenterOAM11gRequest.xml` to serve as a parameter file to the `oamreg` tool.

Copy and paste the example below, and then replace the contents within `$$webtier..$$` with your WebTier host and port IDs, and `$$oam...$$` with the OAM host and administration server port.

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights
reserved.
NAME: OAMRequest.xml - Template (with all options) for OAM Agent
Registration Request file
DESCRIPTION: Modify with specific values and pass file as input to the
tool.
-->
<OAMRegRequest>
   <serverAddress>http://$$oamhost$$:$$oamadminserverport$$</serverAddress>
   <hostIdentifier>$$webtierhost$$_webcenter</hostIdentifier>
   <agentName>$$webtierhost$$_webcenter</agentName>
   <agentBaseUrl>http://$$webtierhost$$:$$webtierport$$</agentBaseUrl>
   <applicationDomain>$$webtierhost$$_webcenter</applicationDomain>
   <autoCreatePolicy>true</autoCreatePolicy>
   <primaryCookieDomain>example.com</primaryCookieDomain>
   <logOutUrls>
        <url>/oamsso/logout.html</url>
    </logOutUrls>
</OAM11GRegRequest>
```

f. Change to the *RREG_Home* directory.

g. Run the following command:

```
RREG_HOME/bin/oamreg.sh inband input/WebCenterOAM11gRequest.xml
```

When prompted for agent credentials enter your OAM administrator credentials.

Enter your WebGate password.

Enter `yes` when asked whether you want to import a URIs file. Specify the full path to the *RREG_HOME*`/input/webcenter.oam.conf` file you copied there earlier.

You should see output like that below indicating that registration has been successful:

```
----------------------------------------
Request summary:
OAM11G Agent Name:example_webcenter
Base URL: http://example.com:8080
URL String:example_webgate
Registering in Mode:inband
Your registration request is being been sent to the Admin server at:
http://example.com:7001
----------------------------------------
Inband registration process completed successfully! Output artifacts are
created in the output folder
```

Note: `ObAcessClient.xml` is generated in the output folder. You will need this file later on after you install WebGate.

h. Change to the *`RREG_HOME`*`/input` directory.

i. From the OAM Console, you should now be able to see the following artifacts:

- 10g WebGate agent named `$$webtierhost$$_webcenter`

- host identifier by the same name

- an application domain with the same name containing authentication and authorization policies which in turn contain protected and public policies

j. Go to **Application Domain> $$webtierhost$$_webcenter > Authentication Policies**. You should be able to see the following policies:

Exclusion Scheme

Protected Resource Policy

Public Resource Policy

WebCenter REST Policy

k. Open the WebCenter REST Policy and change the Authentication Scheme to `BasicScheme` (from the default `LDAPScheme`).

l. Open the Resources tab and search for resources with their Authentication Policy set to `Exclusion Scheme`. You should see the following resources:

```
/rsscrawl*
/rsscrawl/.../*
/sesUserAuth*
/sesUserAuth/.../*
/services-producer/portlets*
/services-producer/portlets/.../*
/wsrp-tools/portlets*
/wsrp-tools/portlets/.../*
```

m. Select the `/rsscrawl*` resource in the search results and click Edit.

n. Change the Protection Level from `Protected` to `Excluded` and click Apply. Note that the resource's authentication policy and authorization policy is removed.

o. Close the Resources tab and repeat the steps for the remaining `Exclusion Scheme` resources.

When you now search for resources with their Authentication Policy set to `Exclusion Scheme` you should see no results.

p. If your installation includes SOA and Content Server deployments, you must update your application policy with SOA and Content Server resources.

Create a policy update file called `WebCenterOAM11gPolicyUpdate.xml` (under *`RREG_HOME`*`/input`) as shown in the example below, replacing the content within `$$webtier..$$` with your Web Tier host and port IDs, and `$$oam...$$` with the OAM host and administration server port:

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
Copyright (c) 2009, 2011, Oracle and/or its affiliates. All rights
reserved.
```

```
NAME: UpdatePolicyRequest.xml - Template for updating application domain
and/or policies without changes to any agent profile
DESCRIPTION: Modify with specific values and pass file as input to the tool
-->
<PolicyRegRequest>


<serverAddress>http://$$oamhost$$:$$oamadminserverport$$</serverAddress>
  <hostIdentifier>$$webtierhost$$_webcenter</hostIdentifier>
  <applicationDomainName>$$webtierhost$$_webcenter</applicationDomainName>

</PolicyRegRequest>
```

q. Run the following command:

```
RREG_HOME/bin/oamreg.sh policyUpdate input/WebCenterOAM11gPolicyUpdate.xml
```

Enter your OAM credentials when prompted.

Enter `yes` when asked whether you want to import a URIs file, and specify *RREG_HOME*`/input/soa.oam.conf`.

Your policy updates with SOA resources.

r. Run the `policyUpdate` command again, this time specifying *RREG_HOME*`/input/oam.conf` to update the policy with Content Server resources. Your policy now contains WebCenter Portal, SOA and Content Server artifacts.

4. **Restart IBM HTTP Server (IHS)**.

5. **Install WebGate 10g**.

OAM 11g can work with both WebGate 10g and 11g but IBM HTTP Server currently only supports WebGate 10g. Therefore, you must download and install WebGate 10g.

a. Download WebGate 10g from Oracle Technology Network. The installable is called **Oracle Access Manager 10g - non OHS11g Webgates and 3rd Party Integrations**.

For Windows, select **Oracle_Access_Manager10_1_4_3_0_Win32_IHS22_ WebGate installer**

For Linux, select **Oracle_Access_Manager10_1_4_3_0_linux_IHS22_WebGate installer**

For Linux, ensure that the GCC libraries are available. If your IHS is 32 bit, choose 32 bit libraries and if your IHS is 64 bit, choose 64 bit libraries.

Also, ensure that User and Group options are correctly set in *IHS_Install_Dir*`/conf/httpd.conf`. Change settings `User nobody` and `Group nobody` to the names of the user and group performing the set up and restart the Web Tier.

b. During installation, specify a location for installing WebGate.

Note: If you ran the installer before and it had failed for some reason, choose a different installation directory.

c. Specify the location of the GCC libraries.

d. Enter the following WebGate details:

WebGate Id: `<agentName>` chosen in the previous step, that is, `$$webtierhost$$_webcenter`

WebGate Password: <password> entered to run oamreg.sh

Access Server Name: oam_server1 (determine this value from OAMConsole)

Access Server HostName: $$oamhost$$

Access Server Port: 5575 (determine this value from OAMConsole)

e. Select to automatically update of httpd.conf and specify the location of httpd.conf from your WebTier. Typically, <webtier>/conf/httpd.conf

f. Finish the wizard.

WebGate successfully installed.

6. **Configure IBM WebSphere for OAM single sign-on.**

Detailed steps are provided in Section 9.9, "Configuring IBM WebSphere for OAM SSO and the IAP". To summarize, you must:

a. Configure a stand alone LDAP registry for OAM in IBM WebSphere.

b. Add and configure a virtual host in IBM WebSphere.

c. Configure IHS reverse proxy in the IBM WebSphere Console.

Note: Ensure you remove all occurrences of <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/"/> and also you make this change each time you generate and propagate the web server plugin.

---

d. Create the Interceptor entry in the IBM WebSphere Console.

e. Configure the OAM TAI configuration file.

If you choose to copy oamtai.xml, ensure that you make the change in the Deployment Manager profile directory as this is the source of truth for other application server profiles.

The values you provide in oamtai.xml are similar to those you provided during WebGate installation. Here's a sample oamtai.xml for reference:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAM-configuration>
   <AAAClientConnect>
       <Parameters>
           <param name = "hostPort" value ="example.com:8080"/>

           <param name = "resource" value ="/Authen/SSOToken"/>
           <param name = "operation" value ="GET"/>
           <param name = "AccessGateName" value ="myWG10g"/>
           <param name = "AccessGatePassword" value ="welcome1"/>
           <param name = "AccessServerHost" value ="oam.example.com"/>
           <param name = "AccessServerPort" value ="5575"/>
           <param name = "AccessServerName" value ="oam_server1"/>
           <param name = "TransportSecurity" value ="open"/>
           <param name = "debug" value ="false"/>
           <param name = "minConn" value ="1"/>
           <param name = "maxConn" value ="1"/>
           <param name = "timeOutForConnPool" value ="30000"/>
<!--
Note:Following parameter is used for Anonymous User Authentication.
```

```
Configure anonymous user value here
-->
          <param name = "Anonymous" value =""/>
<!--
Note:Following parameters are required for Header Based Assertion.
Uncomment it if and only if in case Header based assertion.
1. If you configure the headername here, then the same name will be used
  to configure as return attribute in OAM policy.
   Don't change the assertion type parameter value. Only uncomment the
parameter entry.
2.  If you do not configure the header name here, the default header name
    is "OAM_REMOTE_USER" and the same should be configured in OAM policy.
    Don't change the assertion type parameter value. Only uncomment
parameter entry.
-->
         <param name = "assertionType" value ="HeaderBasedAssertion"/>
         <param name = "customHeaderName" value ="OAM_REMOTE_USER"/>
     </Parameters>
  </AAAClientConnect>
</OAM-configuration>
```

7. Configure logout details.

   a. Configure the single sign-on logout provider.

      Follow steps in Section 9.10.2.2, "Configuring OPSS for SSO Logout with Oracle Access Manager" to update `jps-config.xml`. Ensure that the jps-config.xml you update is from the Deployment Manager profile directory as this is the source of truth for other application server profiles.

   b. Add `oamAuthenProvider.jar` to the classpath.

      Follow steps in Section 9.10.2.3, "Configuring oamAuthenProvider.jar in the IBM WebSphere classpath". Do this for all servers in the install.

   c. Create `logout.html` in WebGate.

      i Navigate to `<WebGate install directory>/access/oamsso`

      ii Create a file called `logout.html` with the following content and configure `SERVER_LOGOUTURL` for your installation:

```
<html>
<head>
<script language="javascript" type="text/javascript">
/////////////////////////////////////////////////////////
//Before using, you need to change the values of:
//a. "oamserverhost" to point to the host where the OAM 11g Server is
running.
//b. "port" to point to the port where the OAM 11g Server is running.
/////////////////////////////////////////////////////////////////////////
var SERVER_LOGOUTURL = "http://example.com:14100/oam/server/logout";
/////////////////////////////////////////////////////////////////////
function delCookie(name,path,domain) {
   var today = new Date();
   var deleteDate = new Date(today.getTime() - 48 * 60 * 60 * 1000); //
minus 2 days
   var cookie = name + "="
            + ((path == null) ? "" : "; path=" + path)
            + ((domain == null) ? "" : "; domain=" + domain)
            + "; expires=" + deleteDate;
   document.cookie = cookie;
}
```

```
function delOblixCookie() {
         // set focus to ok button
      var isNetscape = (document.layers);
  if (isNetscape == false || navigator.appVersion.charAt(0) >= 5) {
    for (var i=0; i<document.links.length; i++) {
      if (document.links[i].href == "javascript:top.close()") {
          document.links[i].focus();
          break;
      }
    }
   }
  delCookie('ObTEMC', '/');
  delCookie('ObSSOCookie', '/');
  delCookie('LtpaToken', '/');
  delCookie('LtpaToken2', '/');
  // in case cookieDomain is configured
  // delete same cookie to all of subdomain
  var subdomain;
  var domain = new String(document.domain);
  var index = domain.indexOf(".");
  while (index > 0) {
     subdomain = domain.substring(index, domain.length);
     if (subdomain.indexOf(".", 1) > 0) {
         delCookie('ObTEMC', '/', subdomain);
         delCookie('ObSSOCookie', '/', subdomain);
         delCookie('LtpaToken', '/', subdomain);
         delCookie('LtpaToken2', '/', subdomain);
     }
     domain = subdomain;
     index = domain.indexOf(".", 1);
   }
}
function handleLogout() {
    //get protocol used at the server (http/https)
    var webServerProtocol = window.location.protocol;
    //get server host:port
    var webServerHostPort = window.location.host;
    //get query string present in this URL
    var origQueryString = window.location.search.substring(1);
    var newQueryString = "";
    //vars to parse the querystring
    var params = new Array();
    var par = new Array();
    var val;

    if (origQueryString != null && origQueryString != "") {
        params = origQueryString.split("&");
        for (var i=0; i<params.length; i++) {
          if (i == 0)
            newQueryString = "?";

        if (i > 0)
            newQueryString = newQueryString + "&";

        par = params[i].split("=");

        //prepare a new query string, if the end_url value needs to be
changed
        newQueryString = newQueryString + (par[0]);
```

```
            newQueryString = newQueryString + "=";
            val = par[1];

            if ("end_url" == par[0]) {
            //check if val (value of end_url) begins with "/" or "%2F" (is it
an URI?)
            if (val.substring(0,1) == "/" || val.substring(0,1) == "%") {
                    //modify the query string now
                    val = webServerProtocol + "//" + webServerHostPort + val;
                }
            }
            newQueryString = newQueryString + val;
        }
        }

        //delete oblix cookies
        delOblixCookie();

        //redirect the user to this URL
        window.location.href = SERVER_LOGOUTURL + newQueryString;
    }
    </script>
    </head>

    <body onLoad="handleLogout();">

    </body>
    </html>
```

d. Update `httpd.conf` in WebTier.

a. Navigate to `<webtier install directory>/conf/httpd.conf`

b. Add the following entries in the `webgate` section

```
Alias /oamsso "<webage-install-dir>/access/oamsso"
```

e. Restart WebTier and all the servers, including the Node Manager in WebSphere.

8. **Configure WebCenter Portal applications to require certificate based authentication**.

   For detailed steps, see Section 5.2.16.2, "Configuring WebCenter Portal Applications for Single Sign-On".

9. **Configure other WebCenter Portal components for single sign-on**:

   - WebCenter Portal: Spaces
   - Discussions server
   - Worklist service
   - RSS news feed service
   - Enterprise Manager
   - Secure Enterprise Search
   - Content Server

   For details steps, see "Additional Single Sign-on Configurations" in Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal.

Note: Since IBM WebSphere only supports one auth-method, once SSO is configured, access through via IBM WebSphere ports results in an error as the required OAM headers are not available. Access to all the applications must be through the Web tier.

5.2.16.2 Configuring WebCenter Portal Applications for Single Sign-On

If you want Spaces or any other application built using WebCenter Portal: Framework, to participate in single sign-on, you must specify `CLIENT-CERT` as the authentication method for Trust Access Interceptors. By default, all WebCenter Portal applications specify `FORM` or `BASIC` as their authentication mechanism.

Unlike Oracle WebLogic Server, IBM WebSphere *does not* support multiple comma separated authentication-method and therefore, you must change the authentication method to CLIENT-CERT for any WebCenter Portal application to participate in single sign-on.

Follow these steps to change authentication method for WebCenter Portal applications:

1. Locate `web.xml` for the WebCenter Portal application.
   - For the Spaces application, go to the machine where IBM WebSphere Application Server is installed and navigate to:

     ```
     PROFILE_DIR/config/cells/
        cellname/applications/webcenter.ear/
        deployments/webcenter/spaces-was.war/WEB-INF/web.xml
     ```

   - For other WebCenter Portal applications, go to the machine where IBM WebSphere Application Server is installed and navigate to the application WAR file. For example:

     ```
     PROFILE_DIR/config/cells/
        cellname/applications/MyPortalApp.ear/
        deployments/MyPortalApp/portal-app.war/WEB-INF/web.xml
     ```

2. Copy `web.xml` to a temporary location.
3. Open `web.xml`in a text editor.
   a. Remove (or comment out) the `<login-config>` section as follows:

      ```
      <login-config>
         <auth-method>FORM</auth-method>
          <form-login-config>
             <form-login-page>/oracle/webcenter/webcenterapp/view/templates/
              publichtml/LoginGateway.jsp</form-login-page>

          <form-error-page>/oracle/webcenter/webcenterapp/view/templates/
              publichtml/LoginGateway.jsp?login_fail=true</form-error-page>
          </form-login-config>
      </login-config>
      ```

   b. Replace with the following `<login-config>` section:

      ```
      <login-config>
        <auth-method>CLIENT-CERT</auth-method>
      </login-config>
      ```

   c. Save the changes.
4. Redeploy the updated `web.xml`file:
   a. Log in to the IBM WebSphere Administrative Console:

```
https://host:port/ibm/console
```

b. Navigate to **Applications > Application Types > WebSphere enterprise applications**.

c. Locate and select the WebCenter Portal application, and then click **Update**.

   To update `web.xml` for the Spaces application, for example, locate the application named `webcenter`.

d. Choose the option **Replace or add a single file**.

e. Specify the path to the `web.xml` file you want to replace. Start the path from the name of the application's archive file (.war):

   war_file_name`/WEB-INF/web.xml`

   For example:

   For the Spaces application, enter: `spaces-was.war/WEB-INF/web.xml`

   For a Framework application, enter: `MyPortalApp.war/WEB-INF/web.xml`

f. Click **Next**.

g. In the section **Specify the path to the file**, enter the full path to the `web.xml` file you updated in step 3.

h. Click **Next**.

i. Click **OK** and **Save Changes**.

   Wait for a couple of minutes for the changes to be propagated.

j. To confirm the change, navigate to the application's deployment descriptor:

   For example, for the Spaces application, navigate to: **WebSphere enterprise Applications > webcenter > Manage Modules > spaces-was.war > View Deployment Descriptor**

5. Restart the WebCenter Portal application.

6. Access the WebCenter Portal application and, if single sign-on is configured, the `web.xml` changes take effect.

7. Repeat similar steps to update any other Web application that participates in single sign-on, for example:

   WebCenter Portal: Spaces install:

   - **Spaces Application:**
     `webcenter/spaces-was.war`
   - **REST API Web Application:**
     `webcenter/webcenter-rest-was.war`
   - **RSS Web Application**:
     `webcenter/webcenter-rss-was.war`
   - **Activity Graph Engines Application**:
     `activitygraph-engines_11.1.1.6.0/activityGraph-engines.war`
   - **Pagelet Producer Admin**:
     `pagelet-producer_11.1.1.6.0/pageletadmin.war`
   - **Pagelet Producer Proxy**:
     `pagelet-producer_11.1.1.6.0/ensembleproxy.war`

- **Services Producer**:
  `services-producer_11.1.1.6.0/services-producer-was.war`
- **Worklist Application**:
  `WebCenterWorklistDetailApp/WebCenterWorklistDetail_was.war`

SOA Suite install:

- **SOA Infra Application:**
  `soa-infra/fabric.war`
- **UMS Application:**
  `usermessagingserver/sdpmessaginguserprefs-ui-web.war`
- **UMS SCA:**
  `usermessagingsca-ui-worklist/sdpmessagingsca-ui-worklist-was.war`
- **Composer Application:**
  `composer/soa-composer-was.war`
- **To Do Task Flow:**
  `DefaultToDoTaskFlow/DefaultToDoTaskFlow.war`

WebCenter Content install:

- **Content Server:**
  `Oracle WebCenter Content-Content Server/cs.war`
- **Inbound Refinery:**
  `Oracle WebCenter Content-Inbound Refinery/ibr.war`

5.2.17 Configuring SSL for WebCenter Portal Applications

Typically, SSL is enabled between the browser and HTTP server. If you need a SSL connection between your IBM HTTP Server and WebSphere nodes (because of your topology/hardened security requirements) or between the browser and WebSphere node directly, you may need to do addition configuration.

This section contains the following subsections:

- Chapter 5.2.17.1, "Obtaining the SSL Port for WebCenter Portal Applications"
- Chapter 5.2.17.2, "Importing SSL Certificates on IBM WebSphere"

5.2.17.1 Obtaining the SSL Port for WebCenter Portal Applications

For SSL between the browser and the WebSphere node, obtain the SSL port as follows:

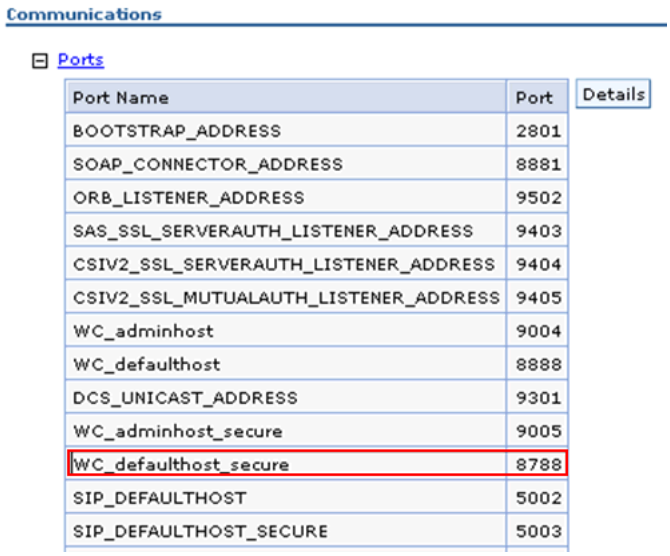
1. Log in to the IBM WebSphere Administrative Console.
2. Navigate to the WebSphere cell on which your WebCenter Portal application is deployed. Select **Application servers> <cell name>**

   For the Space application, for example, navigate to **Application servers> WC_Spaces**.
3. Select **Ports**.

   A list of ports displays. Use the SSL port **WC_defaulthost_Secure** to access your application securely (Figure 5–20).

   For the Space application, for example,
   `https://myhost.com:8788/webcenter`

Figure 5–20 Port Information for WC_Spaces



Port Name	Port
BOOTSTRAP_ADDRESS	2801
SOAP_CONNECTOR_ADDRESS	8881
ORB_LISTENER_ADDRESS	9502
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9403
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9404
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9405
WC_adminhost	9004
WC_defaulthost	8888
DCS_UNICAST_ADDRESS	9301
WC_adminhost_secure	9005
WC_defaulthost_secure	8788
SIP_DEFAULTHOST	5002
SIP_DEFAULTHOST_SECURE	5003

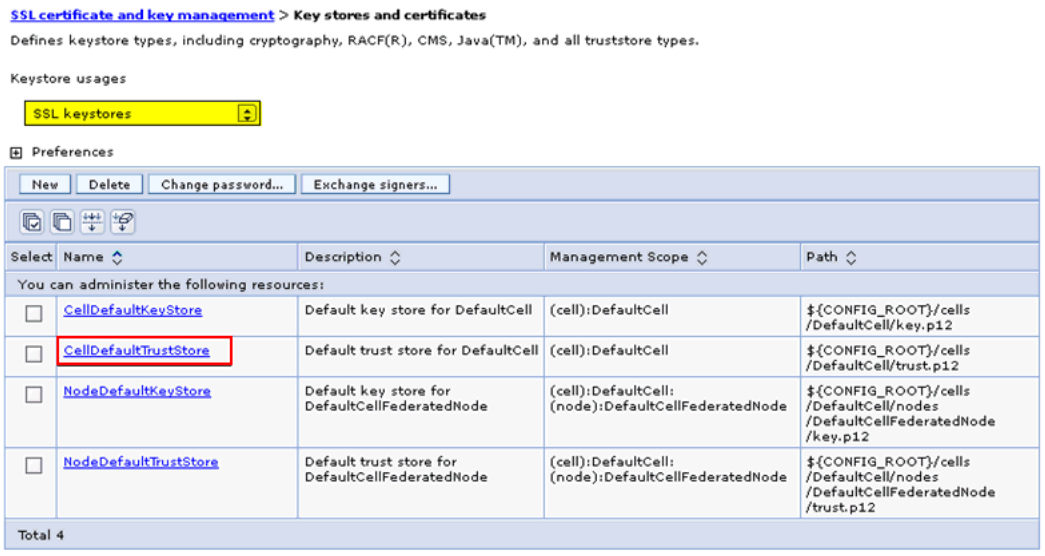
5.2.17.2 Importing SSL Certificates on IBM WebSphere

To import SSL certificates on IBM WebSphere:

1. Log in to the IBM WebSphere Administrative Console.
2. In the navigation panel, expand **Security**, then click **SSL certificate and key management**.
3. Click **Key stores and certificate**s.

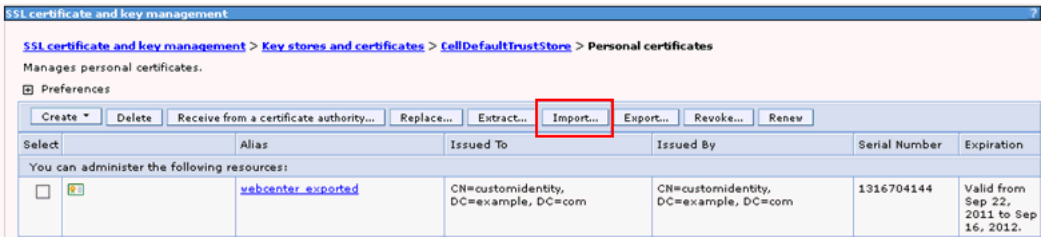
   The **Keystore usages** dropdown should show **SSL keystores** (Figure 5–21).

Figure 5–21 Keystores and Certificates Configuration



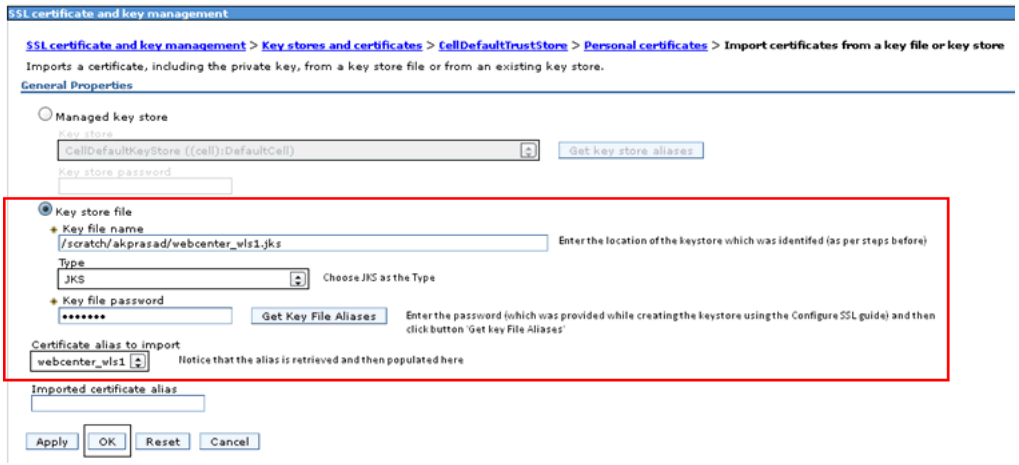
4. Select a trust store (for example, **CellDefaultTrustStore**).
5. Click **Personal Certificate**.
6. Select **Import** (Figure 5–22).

Figure 5–22 CellDefaultTrustStore - Import Personal Certificates



7. Select **Key store file** and specify the location of your keystore (.jks) file (Figure 5–23).

Figure 5–23 Specifying Keystore Location



8. For **Type**, select JKS, and then enter a **Password**.
9. Click **OK** to import the certificates from the keystore.
10. Restart the application server.

5.2.18 Cloning WebCenter Portal Installations on IBM WebSphere

Use the IBM WebSphere Administrative Console to clone WebCenter Portal installations on WebSphere as follows:

1. Log in to the IBM WebSphere Administrative Console.
2. Navigate to **Servers > Server Types > WebSphere application servers.**
3. Create a server template based on the server you want to clone:
   a. Click the **Templates...** button.
   b. On the Server Templates screen, click **New** and select the server for the template.
   c. Click **OK**.
   d. Enter a **Name** for your server template, then click **OK**.
4. Navigate to **Servers > Server Types > WebSphere application servers.**
5. Create an application server based on the template you created in the previous step:

a. Click **New**.

b. Complete **Step 1: Select a node**.

c. For **Step 2: Select a server template**, select the template.

d. Complete Step 3 and Step 4 as required.

The new application server has the same resources as the specified template.

5.2.19 Configuring WebCenter Portal Applications for High Availability on IBM WebSphere

This section describes a typical WebCenter Portal cluster topology and explains some additional set up steps that are required for clustered WebCenter Portal deployments on IBM WebSphere.

This section is not meant to provide comprehensive information for configuring high availability for Oracle WebCenter Portal on IBM WebSphere. For more information about the resources available when configuring high availability on WebSphere, see Section 3.4, "Configuring Oracle Fusion Middleware High Availability on IBM WebSphere".

For an overview of the steps required for setting up high availability for Oracle WebCenter Portal on IBM WebSphere, refer to the following:

1. Install Required WebCenter Portal Components on Both Hosts
2. Configure a New WebSphere Cell on WCPHOST1
3. Federate WCPHOST2 and Configure Cell
4. Configure a Load Balancer
5. Configure Oracle Internet Directory as the LDAP Identity Store
6. Reassociate the Identity Store
7. Configure Distributed Java Object Cache
8. Configure Clustering for Discussions
9. Configure Activity Graph

5.2.19.1 Typical WebCenter Portal Cluster Topology

Figure 5–24 shows a typical cluster set up for a Spaces application deployment.

Figure 5–24 WebCenter Portal Cluster Topology - Spaces Application

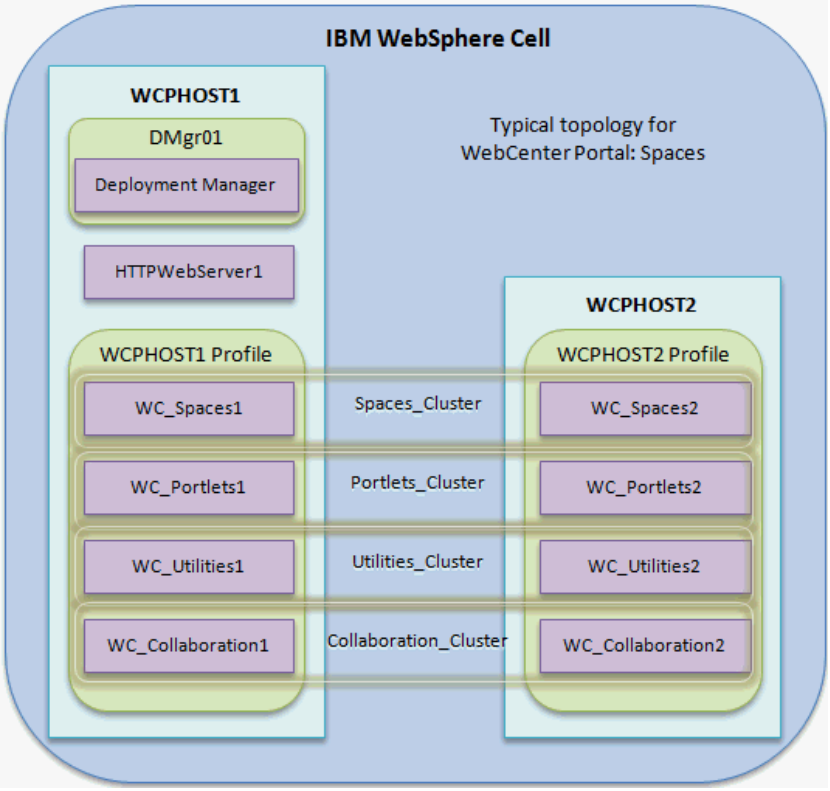
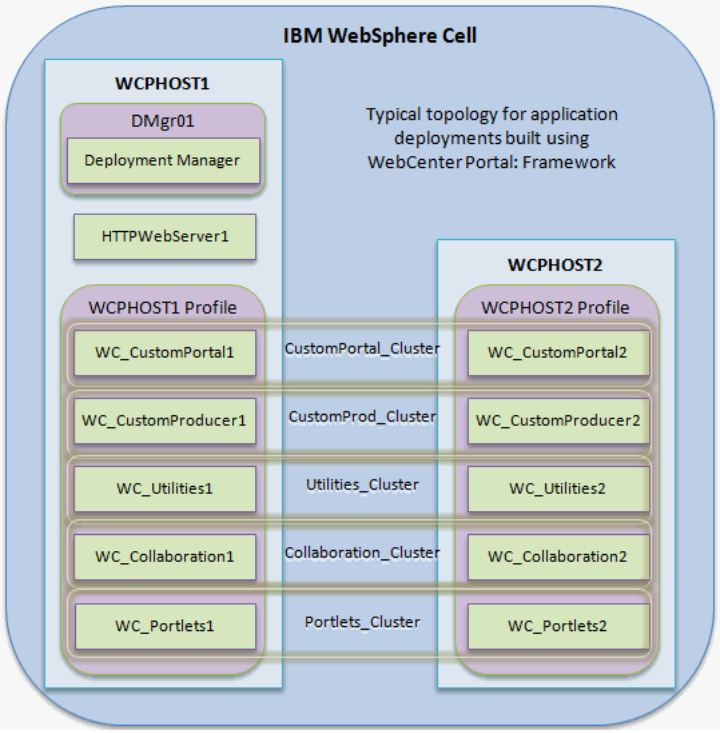


Figure 5–25 shows a typical cluster set up for applications built using WebCenter Portal: Framework, that is, Framework applications and Portlet Producer applications.

Figure 5–25 WebCenter Portal Cluster Topology - Framework and Portlet Producer Applications



5.2.19.2 Install Required WebCenter Portal Components on Both Hosts

In a clustered environment, you must install and configure a suitable database, IBM WebSphere Application Server, Oracle Fusion Middleware (WebCenter Portal, WebCenter Content, SOA Suite), and IBM HTTP Server on *both* hosts. In this section, the hosts are referred to as WCPHOST1 and WCPHOST2.

See also, Section 5.2.1, "Installing WebCenter Portal Products on IBM WebSphere".

5.2.19.3 Configure a New WebSphere Cell on WCPHOST1

On the first WebCenter Portal host (WCPHOST1), create a new WebSphere cell:

1. Launch the Configuration Wizard using:

   ```
   WC_ORACLE_HOME/common/bin/was_config.sh
   ```

2. On the Select Configuration Option page, click **Create and Configure Cell**.
3. On the following screens, select WebCenter Portal products and configure JDBC schemas, as required.

   For details, see "Using the Configuration Wizard" in the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.
4. On the Select Optional Configuration page, click **Application Servers, Clusters and End Points**.
5. On the Configure Cluster page, add the new cluster and the first member, and select **Enabled memory to memory replication**.
6. Finish creating the cell by completing the Configuration Wizard.
7. Start the Deployment Manger:

```
WAS_HOME/profiles/dmg_profile_name/bin/startManager.sh
```

8. Start the node agent on the WCPHOST1 using:

```
WAS_HOME/profiles/profile_name/bin/startNode.sh.
```

5.2.19.4 Federate WCPHOST2 and Configure Cell

If the second WebCenter Portal machine (WCPHOST2), is not yet federated to the cell on WCPHOST1, perform the following steps:

1. Launch the Configuration Wizard using:

```
WC_ORACLE_HOME/common/bin/was_config.sh
```

2. On the Select Configuration Option page, click **Federate Machine and Configure Cell**.
3. Enter Deployment Manager details for WCPHOST1 or the machine where the Deployment Manager is located.

   You can find the machine details at the following location:

```
WAS_HOME/profiles/dmgr_profile_name/logs/AboutThisProfile.txt
```

4. On the Add Products to Cell page, click **Next**.

   You do not need to select any products on this page.
5. On the Select Optional Configuration page, click **Application Servers, Clusters and End Points**.
6. On the Configure Additional Cluster Members page, add the second server for the cluster associated with this node.
7. Finish federating the machine by completing the Configuration Wizard.
8. Start the node agent on Machine2 by opening:

```
WAS_HOME/profiles/profile_name/bin/startNode.sh
```

5.2.19.5 Configure a Load Balancer

Configure an IBM HTTP server for load balancing in a clustered IBM WebSphere environment. For detailed steps, see Section 5.2.15, "Installing and Configuring IBM HTTP Server".

5.2.19.6 Configure Oracle Internet Directory as the LDAP Identity Store

Set up Oracle Internet Directory (OID) as the external LDAP ID store for WebCenter Portal applications:

1. Install and configure Oracle Internet Directory (OID).
2. Configure the LDAP registry.

   Create a properties file for your Oracle Internet Directory LDAP ID store and then run the wsadmin configuration command `Opss.configureIdentityStore`. For detailed steps, see Section 8.1.1, "Configuring a Registry".
3. Perform a full resynchronization for all nodes:
   a. Login to the IBM WebSphere Administrative Console and navigate to the Nodes page (**System administration > Nodes**).
   b. Select all nodes in the cluster and click **Full Resynchronize**.

4. Restart all servers in the cluster.

5.2.19.7 Reassociate the Identity Store

Perform the following steps to reassociate the identity store:

1. Connect to the Dmgr server using wsadmin:

```
WC_HOME/common/bin/wsadmin.sh -conntype SOAP -user <admin username> -password
<password> -port <admin SOAP port> -host <Dmgr host>
```

2. Run the security store reassociate wsadmin command:

```
Opss.reassociateSecurityStore(domain="Cell_WebSphere",servertype="OID",
ldapurl="ldap://<host>:<port>", jpsroot="<jpsroot>",
admin="<admin username>", password="<admin password>")
```

3. Perform a full resynchronization for all nodes:
   a. Login to the IBM WebSphere Administrative Console and navigate to the Nodes page (**System administration > Nodes**).
   b. Select all nodes in the cluster and click **Full Resynchronize**.
4. Restart all servers in the cluster.

5.2.19.8 Configure Distributed Java Object Cache

For details how to configure the distributed Java Object Cache (JOC), see Section 3.4.2, "Configuring Java Object Cache for Oracle Fusion Middleware on IBM WebSphere".

5.2.19.9 Configure Clustering for Discussions

Configure clustering options for Discussions, deployed on the WC_Collaboration servers:

1. Create an Admin user for the Discussions server:

   For details, see Section 5.2.8, "Configuring an Admin User for the Discussions Server".
2. Restart all WC_Collaboration servers in the cluster.
3. For each server in the WC_Collaboration cluster, configure unicast cluster communication:
   a. Log into the IBM WebSphere Administrative Console.
   b. Expand **Servers**, expand **Server Types**, then click **WebSphere application servers**.
   c. Click **WC_Collaboration**, the server on which the Discussions application is deployed.
   d. Under Server Infrastructure, expand **Java and Process Management**, then click **Process definition**.
   e. On the process definition page, under Additional Properties, click **Java Virtual Machine**.
   f. On the Java virtual machine page, under Additional Properties, click **Custom properties**.
   g. Create the following variables. Click **New**, enter the name, enter the value, then click **OK**.

Name: **tangosol.coherence.wka1**, Value=**WCPHOST1**

Name: **tangosol.coherence.wka2**, Value=**WCPHOST2**

Name: **tangosol.coherence.localhost**, Value=**WCPHOST1**

Name: **tangosol.coherence.wka1.port**, Value=**8089**

Name: **tangosol.coherence.wka2.port**, Value=**8089**

Name: **tangosol.coherence.localport**, Value=**8089**

4. Repeat step 3 for WC_Collaboration2, swapping WCPHOST1 for WCPHOST2, and WCHost2 for WCHost1:

   Name: **tangosol.coherence.wka1**, Value=**WCPHOST1**

   Name: **tangosol.coherence.wka2**, Value=**WCPHOST2**

   Name: **tangosol.coherence.localhost**, Value=**WCPHOST1**

   Name: **tangosol.coherence.wka1.port**, Value=**8089**

   Name: **tangosol.coherence.wka2.port**, Value=**8089**

   Name: **tangosol.coherence.localport**, Value=**8089**

5. Restart all WC_Collaboration servers in the cluster.

5.2.19.10 Configure Activity Graph

The Activity Graph application (`activitygraph-engines`) cannot be targeted to a cluster. As IBM WebSphere does not allow you to target an application to a server that is part of a cluster, you must create a new server for the Activity Graph application:

1. Create a server template based on the **WC_Utilities** server
   a. Log into the IBM Admin Console.
   b. Expand **Servers**, expand **Server Types**, then click **WebSphere application servers**.
   c. Click **Templates**.
   d. Click **New**.
   e. Select **WC_Utilities** as the server for the template, then click **OK**.
   f. Enter the name for the server template, then click **OK**.
2. Create the new server based on the newly created server template:
   a. Expand **Servers**, expand **Server Types**, then click **WebSphere application servers**.
   b. Click **New**.
   c. Select the node where this server is located, enter the server name, then click **Next**.
   d. Select the template you just created, then click **Next**.
   e. Select **Generate Unique Ports**, then click **Next**.
   f. Click **Finish**.
3. Re-target the application to the new server:
   a. Expand **Applications**, expand **Application Types**, then click **WebSphere enterprise applications**.

b. Click **activitygraph-engines_11.1.1.4.0**.

c. Under Modules, click **Manage Modules**.

d. Select all modules, select the target server you created in step 2, and click **Apply**.

e. Click **OK**.

5.3 Differences Developing and Deploying WebCenter Portal Applications on IBM WebSphere

The following sections describe differences and restrictions when developing and deploying WebCenter Portal applications on IBM WebSphere:

- Configuring a WebSphere Application Server Connection in JDeveloper
- Deploying WebCenter Portal Applications on IBM WebSphere Directly from JDeveloper
- Targeting Application EAR and WAR Files for IBM WebSphere Deployment
- Deploying WebCenter Portal Application EARs using WebSphere Console and wsadmin
- Securing a Framework Application Connection to IMAP and SMTP with SSL
- Using the Deploy and Configure Script for WebCenter Portal Applications Deployed on WebSphere
- Creating SQL Data Controls for Applications Deployed on WebSphere Administration Server

5.3.1 Configuring a WebSphere Application Server Connection in JDeveloper

If you want to deploy a WebCenter Portal application to an IBM WebSphere Server that resides outside JDeveloper, you must ensure that the target server is up and running with the required libraries, and then you must.set up a connection to the target WebSphere server.

During application server connection creation, you are prompted for configuration information on several wizard pages. Table 5–4 describes where to find this information in the IBM WebSphere Administrative Console for which you are prompted.

Table 5–4 Location of Application Server Connection Configuration Details

Connection Wizard Fields	For IBM WebSphere Application Server - 7.0, Select...	For IBM WebSphere Application Server - Network Deployment (ND), Select...
Configuration Page		
■ **SOAP Connector Port**	**System administration > Deployment manager > Configuration > Ports > SOAP_CONNECTOR_ADDRESS**	**System administration > Deployment manager > Configuration > Ports > SOAP_CONNECTOR_ADDRESS**

Table 5–4 (Cont.) Location of Application Server Connection Configuration Details

Connection Wizard Fields	For IBM WebSphere Application Server - 7.0, Select...	For IBM WebSphere Application Server - Network Deployment (ND), Select...
▪ Server Name	System administration > Deployment manager > Configuration > Name	Servers > Server Types > WebSphere Application Servers > *Your_Server_Name* > Configuration > Name
▪ Target Node	System administration > Deployment manager > Runtime > Node name	Servers > Server Types > WebSphere Application Servers > *Your_Server_Name* > Runtime > Node name
▪ Target Cell	System administration > Deployment manager > Runtime > Cell name	Servers > Server Types > WebSphere Application Servers > *Your_Server_Name* > Runtime > Cell name

For more information, see Section 4.2.4, "Creating an Application Server Connection" and "Connecting and Deploying Java EE Applications to Application Servers" in *Oracle Fusion Middleware User's Guide for Oracle JDeveloper*.

5.3.2 Deploying WebCenter Portal Applications on IBM WebSphere Directly from JDeveloper

Deploying WebCenter Portal applications directly from Oracle JDeveloper to IBM WebSphere Server is largely the same as described in *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*.

The differences are as follows:

- **Database connection configuration for seeded data sources is different on IBM WebSphere.** For details, see:
  - Creating Database Connections for Seeded Data Sources on Out-of-the-Box Server
  - Creating Database Connections for Seeded Data Sources on Other Target Servers
- **Database connection configuration for custom data sources is different on IBM WebSphere.** For details, see:
  - Creating Database Connections to Custom Data Sources

  Note: Custom data sources *are not* automatically created when you deploy WebCenter Portal applications to a WebSphere Application Server through JDeveloper or Fusion Middleware Control.
- The **Deploy using SSL** check box does not appear on the Create Application Server Connection wizard page. See "Deploying WebCenter Portal Applications Using SSL".
- **Deployed applications do not start automatically after deployment.** You have to start the WebCenter Portal application manually using the console. See "Deploying and Redeploying WebCenter Portal Applications From JDeveloper".

5.3.2.1 Creating Database Connections for Seeded Data Sources on Out-of-the-Box Server

Servers created using **WC_CustomPortal** and **WC_CustomServicesProducer** templates automatically come pre-seeded with two data sources (**WebCenter** and **Activities**):

Data Source	Data Source Name	JNDI Name
WC_CustomPortal Template		
WebCenter	webcenter/CustomPortalDS	jdbc/webcenter/CustomPortalDS
Activities	activities/CustomPortalDS	jdbc/activities/CustomPortalDS
WC_CustomServicesProducer Template		
WebCenter	webcenter/CustomServicesProducerDS	jdbc/webcenter/CustomServicesProducerDS
Activities	activities/CustomServicesProducerDS	jdbc/activities/CustomServicesProducerDS

No additional configuration is required if you want to deploy a working WebCenter Portal application directly to a WC_CustomPortal or WC_CustomServicesProducer server, through JDeveloper.

Optionally, if you plan to test the application in JDeveloper's embedded *WebLogic Server*, you must manually create the relevant database connections, ensuring that the database connection names map to the data sources in the target server as follows:

Data Source	Database Connection Name		
WebCenter	webcenter/CustomPortal	or	webcenter/CustomServicesProducer
Activities	activities/CustomPortal	or	activities/CustomServicesProducer

When the bindings are generated in the deployment descriptor, the above connection names are prefixed with the sting "`jdbc/`" and appended with the string "`DS`".

To enable application testing in the embedded *WebLogic Server*, create database connections for seeded data sources as follows (this example illustrates deployment to a WC_CustomPortal server):

1. In JDeveloper, create a database connection.

   For general steps, see "Setting Up a Database Connection" in *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*.

2. To connect to the WebCenter Portal database, ensure that:
   - Connection Name = webcenter/Custom Portal (Figure 5–26)
   - Associate to Data Source = WebCenterDS (Figure 5–27)

Figure 5–26 Database Connection Name - WebCenter/CustomPortal

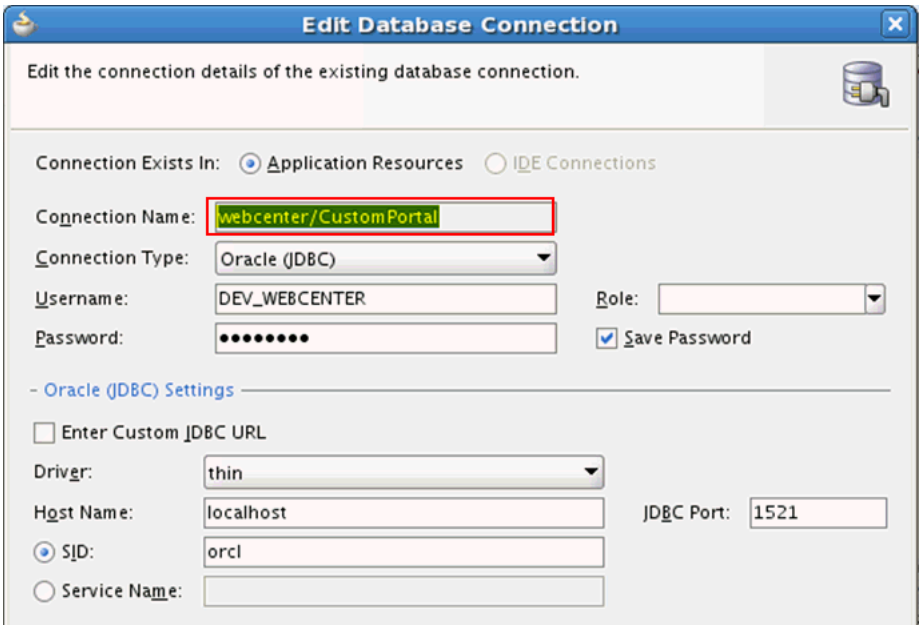
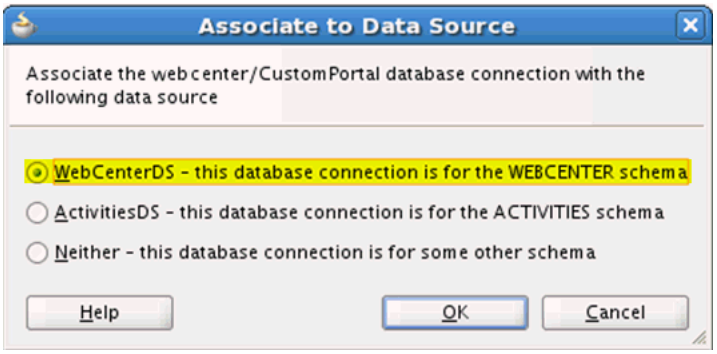


Figure 5–27 Associate webcenter/CustomPortal Connection to WebCenterDS



3. To connect to the Activities database, ensure that:
   - Connection Name = activities/Custom Portal (Figure 5–28)
   - Associate to Data Source = ActivitiesDS (Figure 5–29)

Figure 5–28 Database Connection Name - activities/CustomPortal

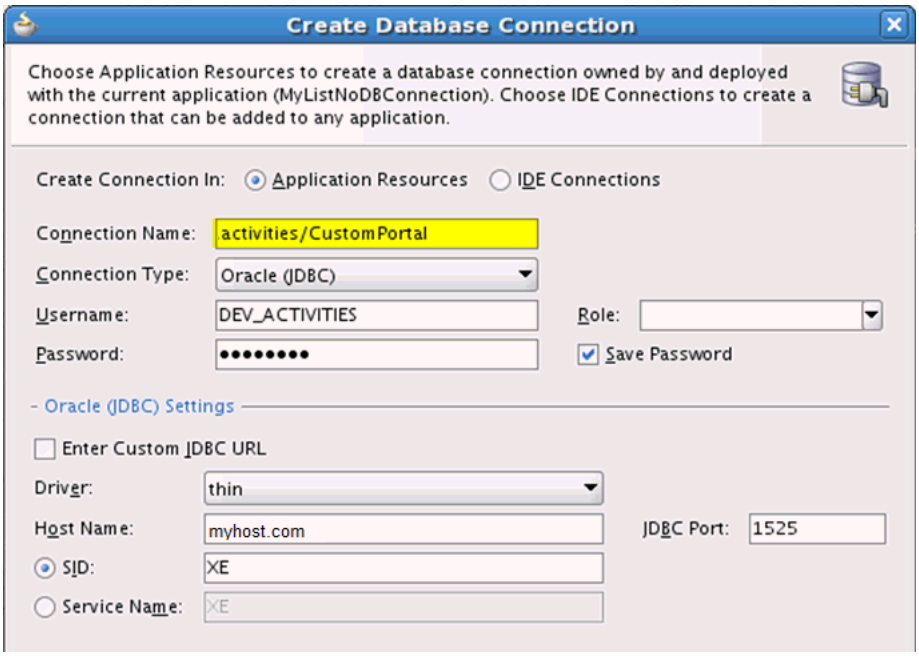
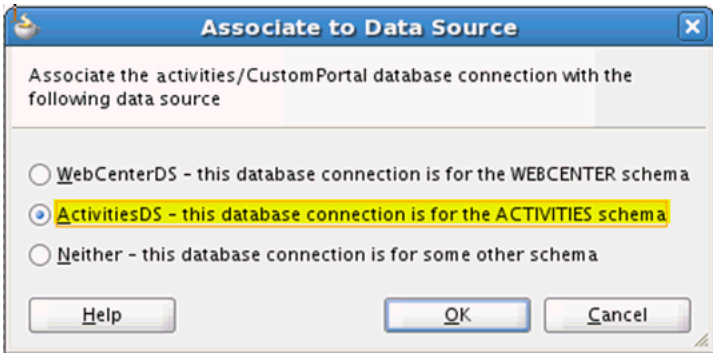
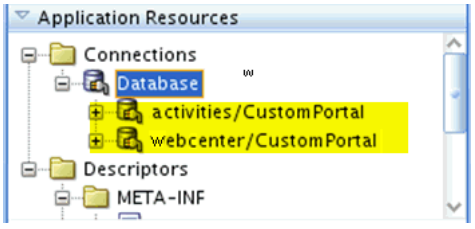


Figure 5–29 Associate activities/CustomPortal Connection to ActivitiesDS



The database connections appear in the navigator as follows:

Figure 5–30 Application Navigator - Database Connections to Seeded Data Sources



5.3.2.2 Creating Database Connections for Seeded Data Sources on Other Target Servers

Oracle recommends that WebCenter Portal applications built using WebCenter Portal: Framework are deployed on servers created using the **WC_CustomPortal** template (and come pre-seeded with **WebCenter** and **Activities** data sources). However, if you must deploy your WebCenter Portal application to a different target server (such as

WC_CustomServicesProducer) and want to use the WebCenter or Activities data sources that have been created or pre-exist on the target server, you must manually create database connections to the WebCenter and Activities data sources.

To ensure that the correct bindings are generated for the data sources in the target server, the names of the database connections must match up with the existing JNDI name (if any). When the bindings are generated in the deployment descriptor, connection names are prefixed with the sting "`jdbc/`" and appended with the string "`DS`". For example:

Data Source	JNDI Name	Database Connection Name
WebCenter	jdbc/MyWebCenterDS	MyWebCenter
Activities	jdbc/ActivitiesDS	MyActivities

Failure to create a database connection results in run time failures with log entries such as:

```
Caused by: javax.naming.NameNotFoundException:
Context: Cell1/nodes/Server1Node/servers/WC_Spaces,
name: jdbc/webcenter/CustomPortalDS:
First component in name webcenter/CustomPortalDS not found.
Root exception is org.omg.CosNaming.NamingContextPackage.NotFound:
IDL:omg.org/CosNaming/NamingContext/NotFound:1.0]
```

Failure to create binding entries that do not match an existing data source JNDI name result in run time failures with log entries such as:

```
Caused by: javax.naming.NameNotFoundException:
Context: Cell1/nodes/Server1Node/servers/WC_Spaces,
name: jdbc/MyWebCenterDS:
First component in name MyWebCenterDS not found.
[Root exception is org.omg.CosNaming.NamingContextPackage.NotFound:
IDL:omg.org/CosNaming/NamingContext/NotFound:1.0]
```

To create database connections for seeded data sources on a server *other than* WC_CustomPortal:

1. In JDeveloper, create a database connection.

   For general steps, see "Setting Up a Database Connection" in *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*.

2. To connect to the WebCenter Portal database, ensure that:

   - Connection Name = <matches the JNDI name>

     For example, if the JNDI name is `jdbc/MyWebCenterDS`, the connection name must be `MyWebCenter` (Figure 5–31)

   - Associate to Data Source = WebCenterDS (Figure 5–32)

Figure 5–31 Database Connection Name - MyWebCenter

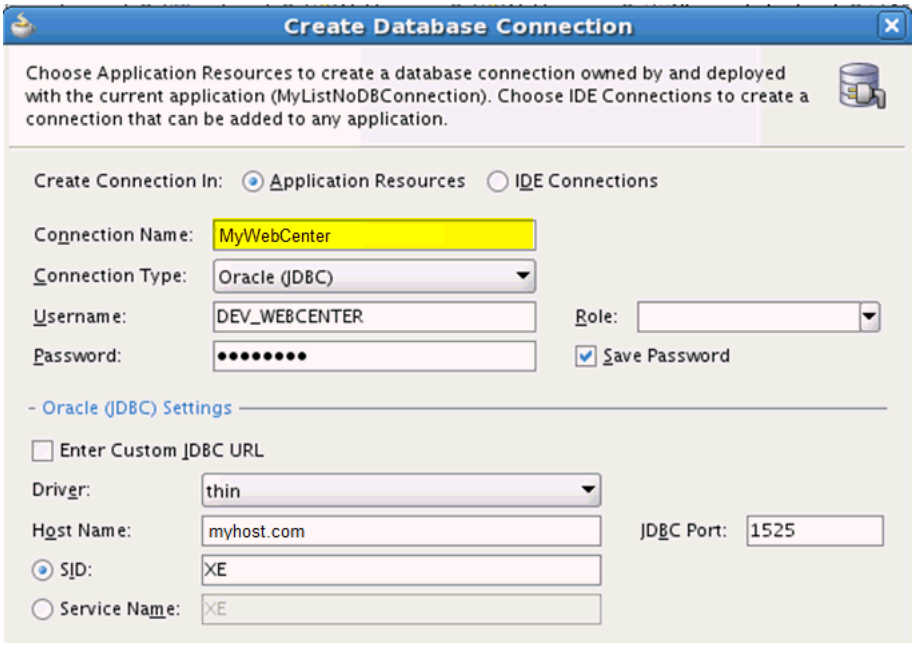
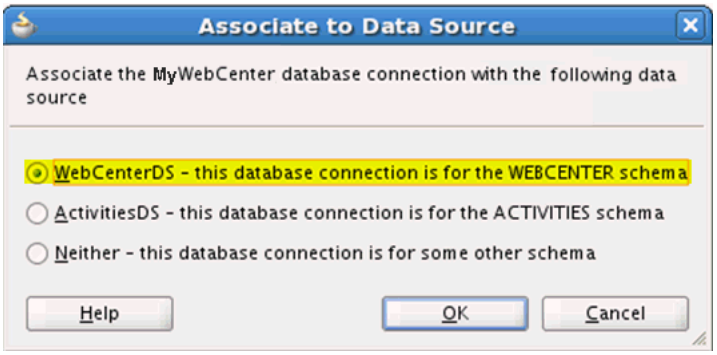


Figure 5–32 Associate MyWebCenter Connection to WebCenterDS



3. To connect to the Activities database, ensure that:
   - Connection Name = <matches the JNDI name>

     For example, if the JNDI name is `jdbc/MyActivitiesDS`, the connection name must be `MyActivities` (Figure 5–33)
   - Associate to Data Source = ActivitiesDS (Figure 5–34)

Figure 5–33 Database Connection Name - MyActivities

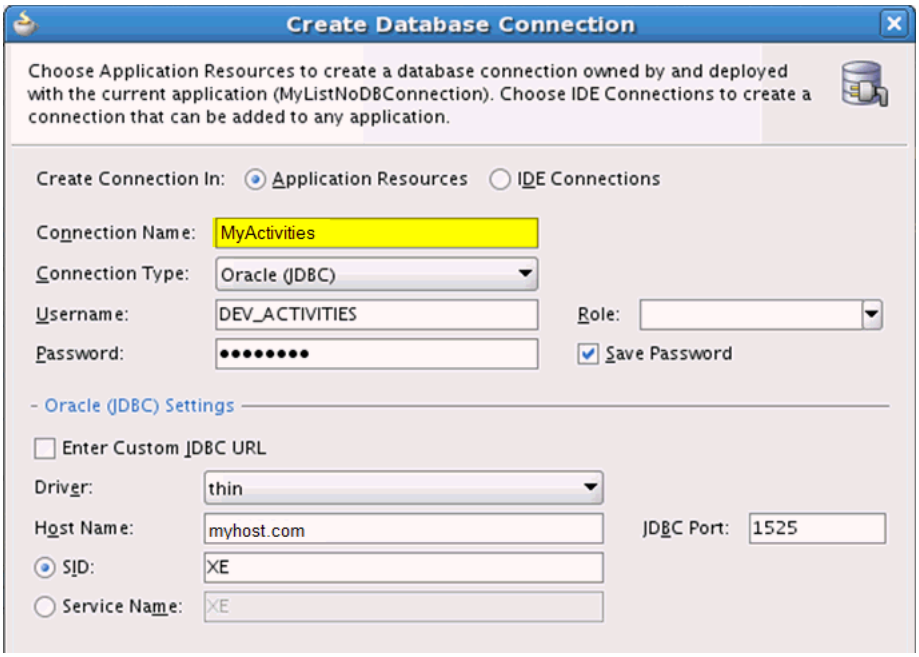
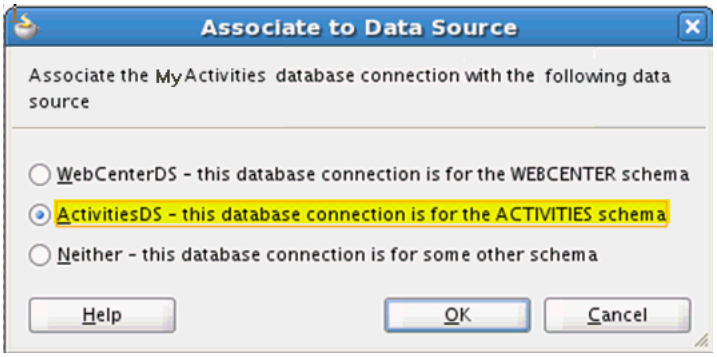
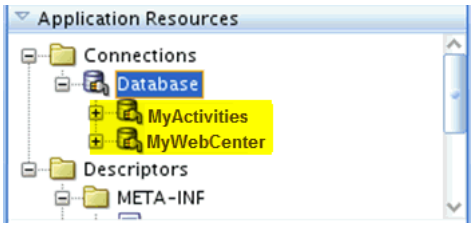


Figure 5–34 Associate MyActivities Connection to ActivitiesDS



The database connections appear in the navigator as follows:

Figure 5–35 Application Navigator - Database Connections to Data Sources



5.3.2.3 Creating Database Connections to Custom Data Sources

Custom data sources are not automatically created when you deploy WebCenter Portal applications built using WebCenter Portal: Framework to an IBM WebSphere Application Server through JDeveloper and Fusion Middleware Control. If you want to use data sources other than those provided by the template (*WebCenter* and

Activities), you must create the custom data sources manually using the IBM WebSphere Administrative Console.

Firstly, at design-time, you must create a database connection and map it to the WebCenter or Activities schema. Once complete, you can note down the associated JNDI name that the application will use post deployment and create a data source that maps to that JNDI name.

Mapped Data Source	Database Connection Name	JNDI Name
WebCenter or Activities	<DatabaseConnectionName>	jdbc/<DatabaseConnectionName>DS
For example:		
WebCenterDS	MyLists	jdbc/MyListsDS

To create the database connection and verify the JNDI name:

1. In JDeveloper, create a database connection.

   For general steps, see "Setting Up a Database Connection" in *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*.

2. To connect to a custom data source instead of the default WebCenter Portal database, ensure that:
   - Connection Name = <*any name*> For example, MyLists (Figure 5–36)
   - Associate to Data Source = WebCenterDS or ActivitiesDS (Figure 5–37)

Figure 5–36 Custom Database Connection Name - MyLists

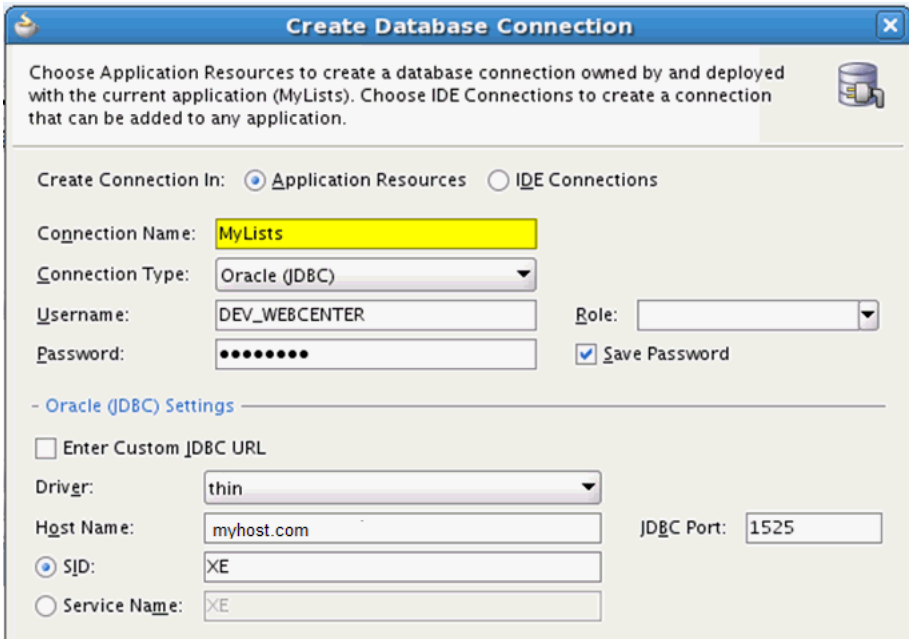
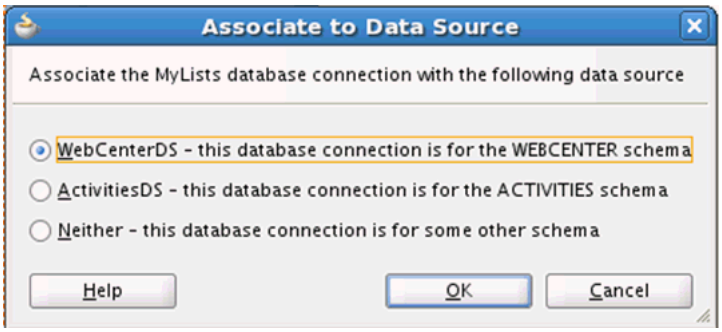


Figure 5–37 Associate Custom Database Connection to WebCenterDS

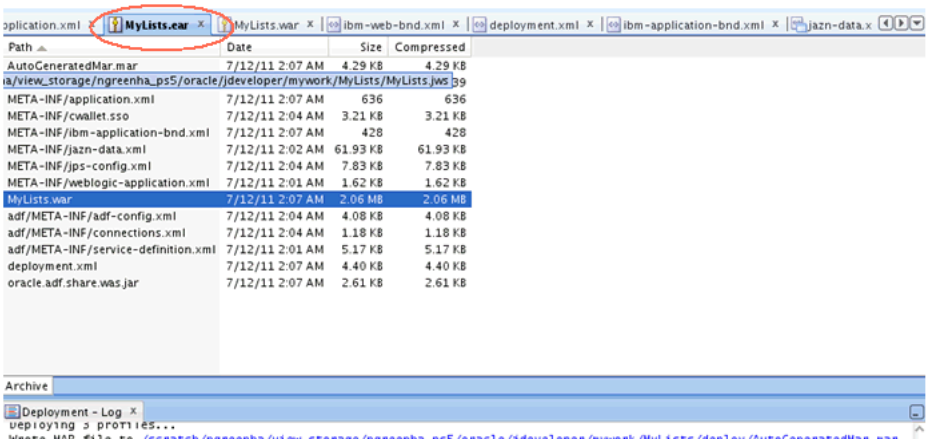


This step ensures that when you deploy the application, deployment descriptor updates map the MyLists data source name to all usages of the WebCenter schema for this application. For instance, in this example, the MyLists data source specifies an alternative back-end repository for the List service.

Similarly, you can specify an alternative data source for all usages of the Activities schema.

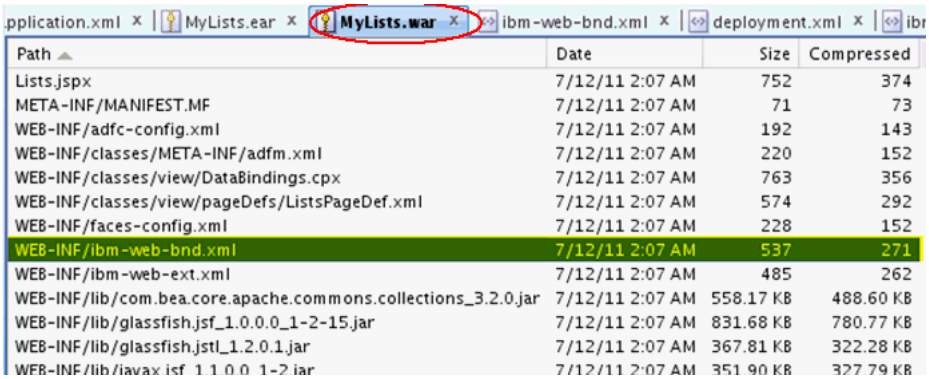
3. To examine the content of the EAR file and validate the mapping, deploy the application to the file system (ensuring the target platform is specified as IBM WebSphere), and then click the EAR file in the JDeveloper log file (Figure 5–38)

Figure 5–38 EAR File Deployed to IBM WebSphere



4. Select the WAR file, and navigate to `WEB-INF/ibm-web-bnd.xml` (Figure 5–39).

Figure 5–39 WAR File Deployed to IBM WebSphere



5. Open `WEB-INF/ibm-web-bnd.xml` to verify that the binding entry maps the MyLists entry to WebCenterDS usages in the application (Figure 5–40).

   Note: The binding-name is `jdbc/MyListsDS`—the database connection name prefixed with "`jdbc`" and appended with "`DS`". This information must be used when you create the data source for the target server, using the IBM WebSphere Administrative Console.

Figure 5–40 ibm-web-bnd.xml Deployed to IBM WebSphere

To create a data source using the IBM WebSphere Administrative Console:

1. Log in to the IBM WebSphere Administrative Console.

   `https://host:port/ibm/console`

2. Navigate to **Guided Activities > Connecting to a database**.

   See also, Section 3.2.6, "Creating a Data Source in an IBM WebSphere Cell".

3. Configure credentials for secure database access (Figure 5–41).

Figure 5–41 Configure Credentials for Secure Database Access

4. Configure a JBDC provider (Figure 5–42).

Figure 5–42 Configure a JBDC Provider

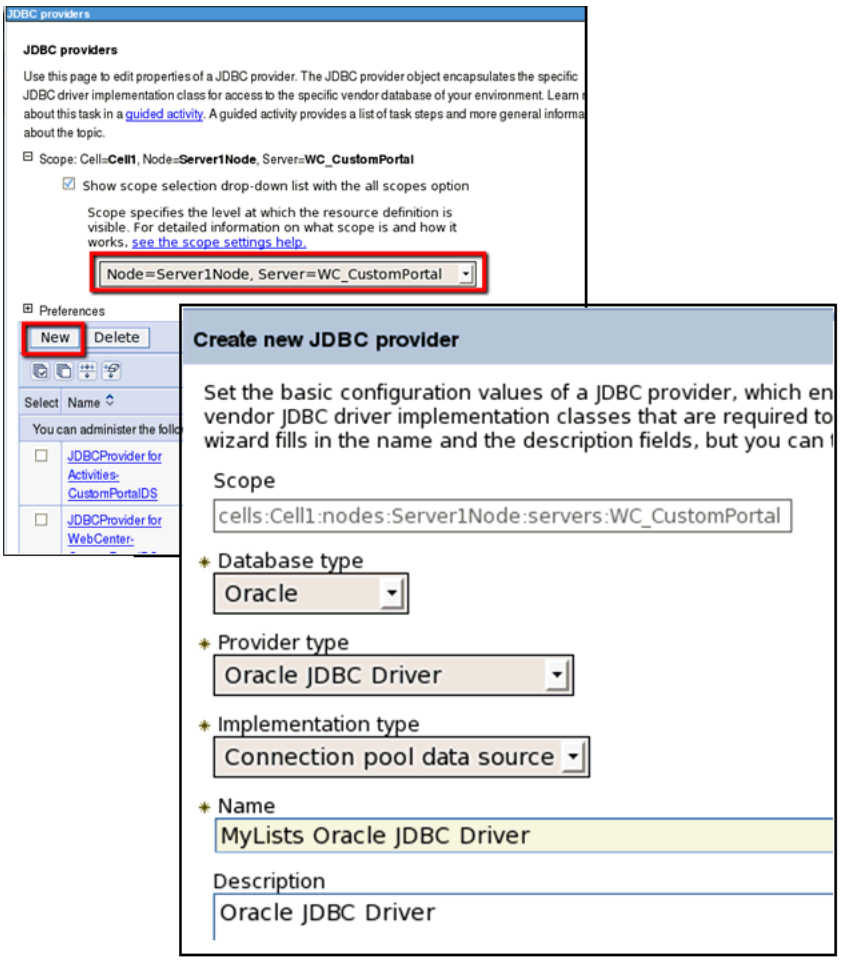
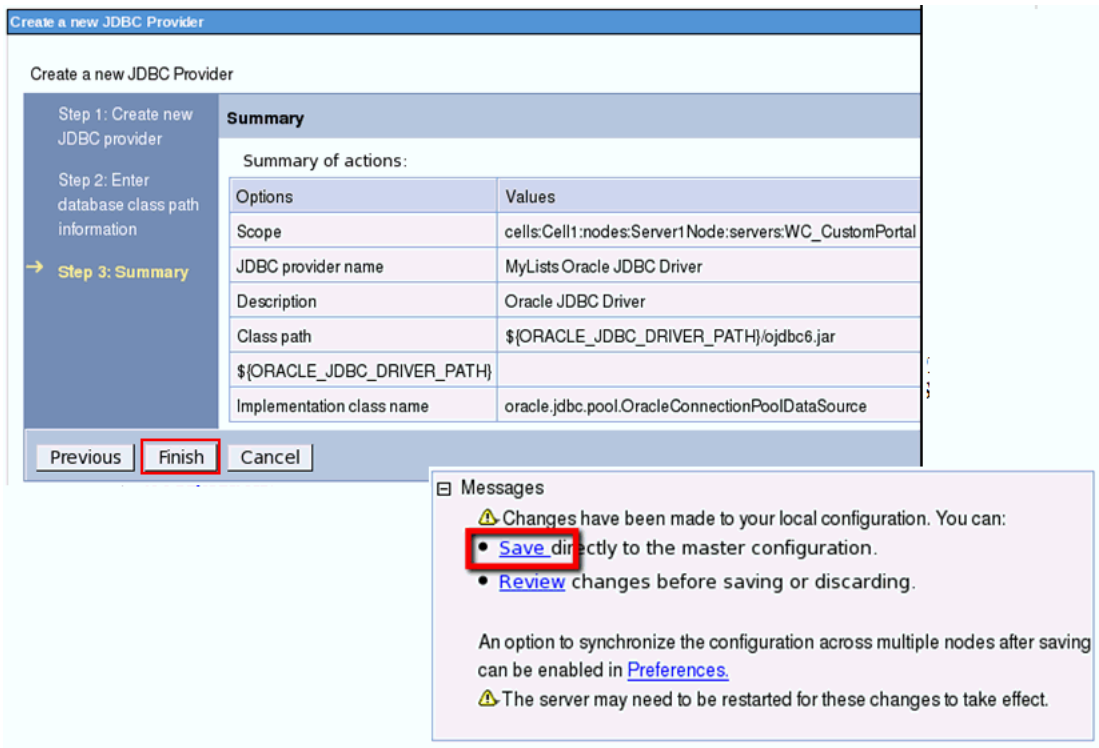


Figure 5–43 Create and Save the JBDC Provider

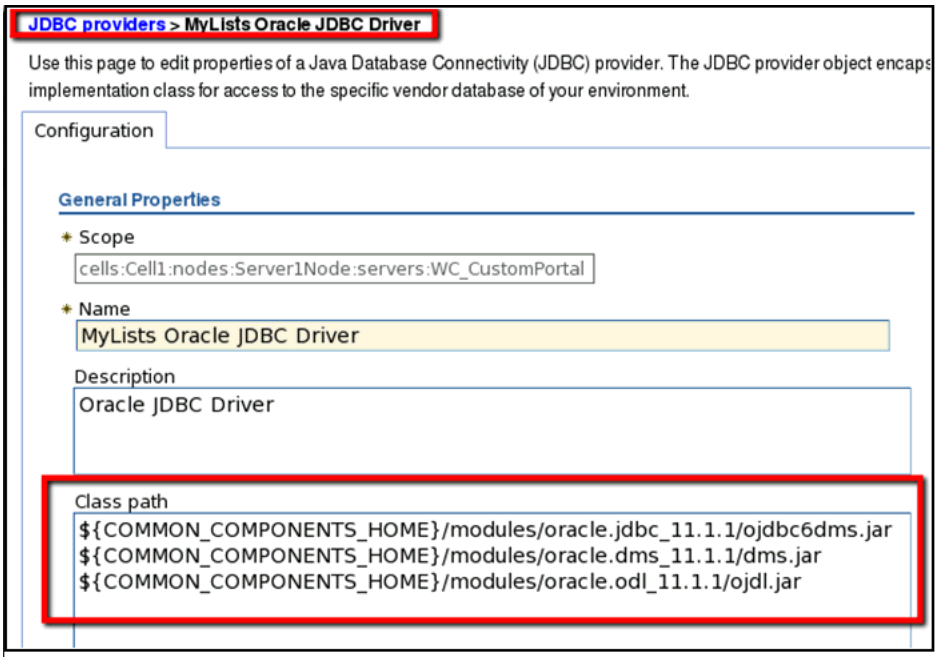


5. Modify the JDBC provider to use the latest Oracle database classes (Figure 5–44):

```
${COMMON_COMPONENTS_HOME}/modules/oracle.jdbc_11.1.1/ojdbc6dms.jar
${COMMON_COMPONENTS_HOME}/modules/oracle.dms_11.1.1/dms.jar
${COMMON_COMPONENTS_HOME}/modules/oracle.odl_11.1.1/ojdl.jar
```

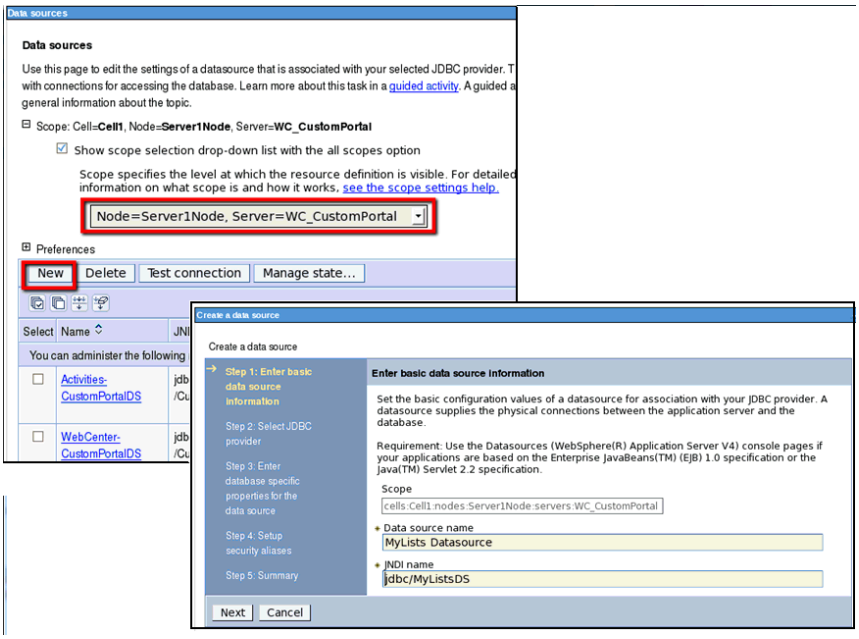
   and save your changes to the master configuration.

Figure 5–44 Add the Latest Oracle Database Classes



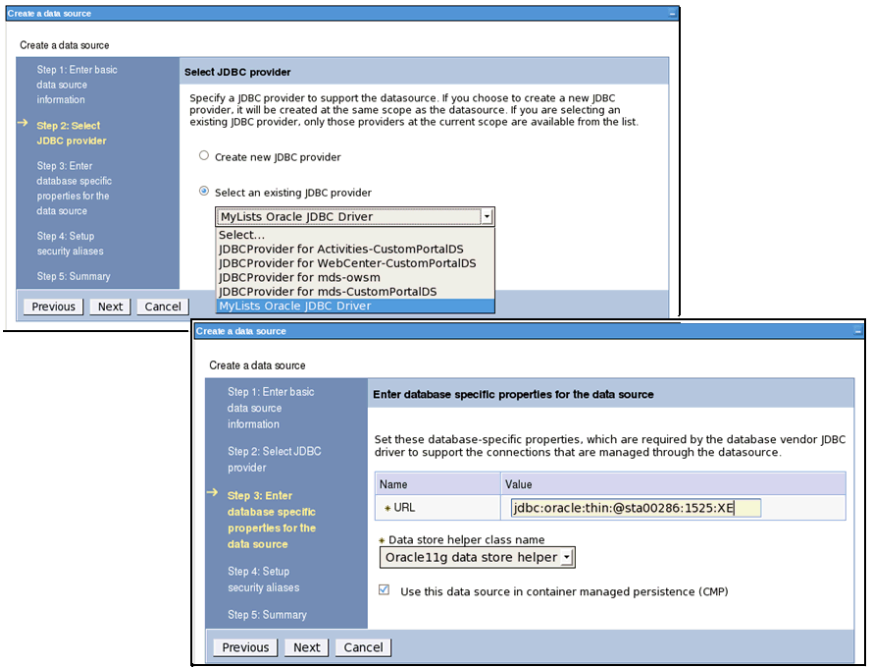
6. Skip the step "Configure WebSphere variables".
7. Configure a data source:
   a. Enter the **JNDI Name** exactly as it appears in the application bindings (Figure 5–45). For example: `jdbc/MyListsDS`

Figure 5–45 Configure the Data Source



   b. Click **Next**. Select the **JBDC Provider** you created earlier and enter the JBDC **URL** to the database connection you created in JDeveloper (Figure 5–46).

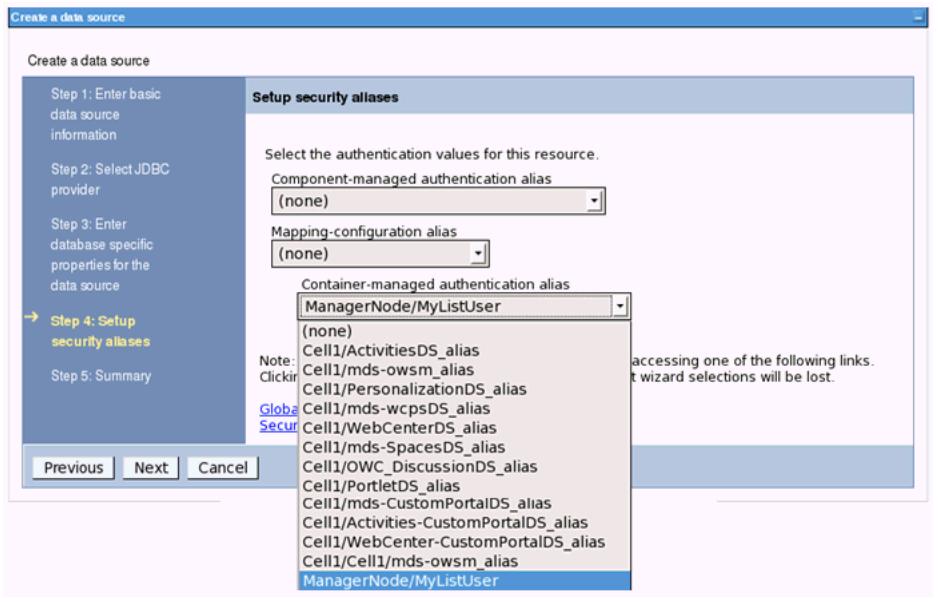
Figure 5–46 Configure the JDBC Provider



c. Click **Next**. For **Container-managed authentication alias**, select the alias created earlier, for example MyListUser, and leave all the other fields blank (Figure 5–47).

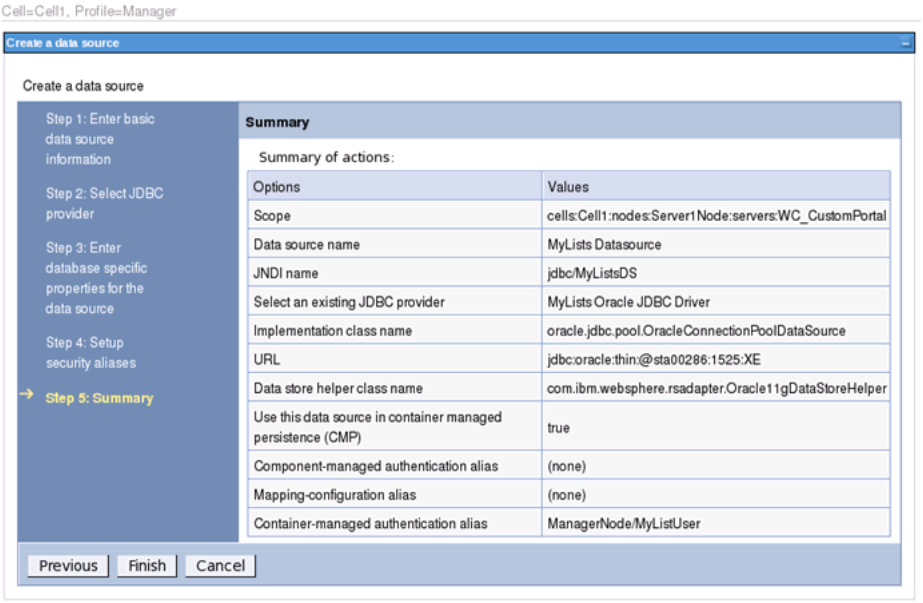
Note: Component managed authentication is required if you plan to build SQL data controls. For details, see Section 5.3.7, "Creating SQL Data Controls for Applications Deployed on WebSphere Administration Server".

Figure 5–47 Select the Data Source Security Alias



8. Click **Next**. Confirm the changes and click **Finish** (Figure 5–48).

Figure 5–48 Save Data Source Configuration



9. Restart the servers to effect the new authentication alias *MyListUser*.
10. Test the data source connection.
11. Deploy the application and it verify that it uses the new custom data source.

5.3.2.4 Deploying WebCenter Portal Applications Using SSL

When you deploy WebCenter Portal applications to IBM WebSphere from JDeveloper, a **Deploy using SSL** check box displays on the deployment wizard. This differs from Oracle WebLogic Server deployment, where the **Deploy using SSL** check box instead appears on the Create Application Server Connection wizard (Configuration page).

Table 5–5 describes what occurs when you select this check box during IBM WebSphere Server deployment.

Table 5–5 Deployment to HTTPS and HTTP Servers

If This Checkbox Is...	Then...
Selected	An HTTPS server URL must exist to deploy the application with SSL. If the server only has an HTTP URL, deployment fails.
Not selected	An HTTP server URL must exist to deploy to a non-SSL environment. Otherwise, deployment fails. If the server has both HTTPS and HTTP URLs, deployment occurs through a non-SSL connection. This enables you to force a non-SSL deployment from Oracle JDeveloper, even though the server is SSL-enabled.

5.3.2.5 Deploying and Redeploying WebCenter Portal Applications From JDeveloper

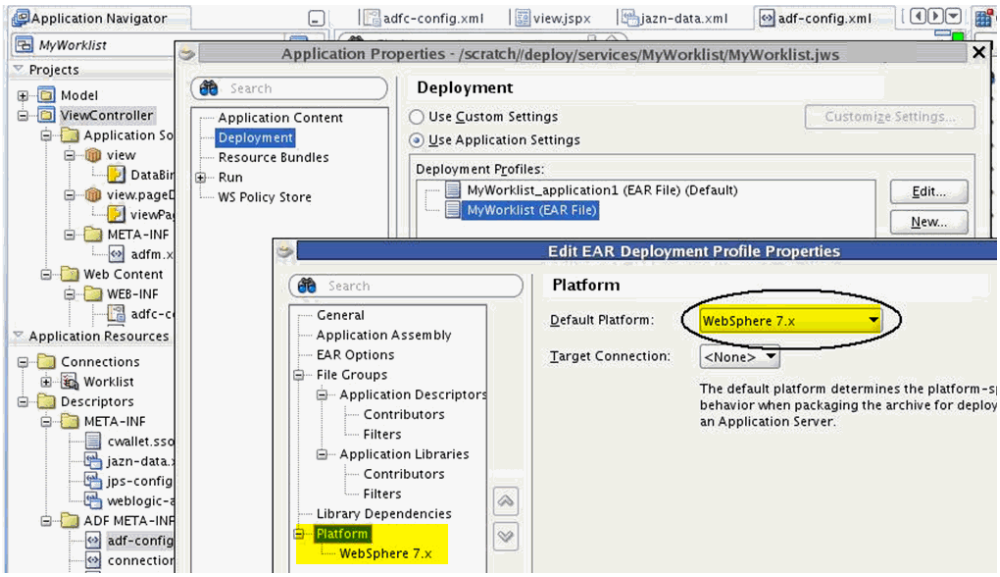
Applications do not start automatically after deployment or redeployment from JDeveloper. You have to start the WebCenter Portal application manually using IBM

WebSphere Administrative Console or wsadmin commands. For details, see Deploying WebCenter Portal Application EARs using WebSphere Admin Console and Deploying WebCenter Portal Application EARs using wsadmin Commands.

5.3.3 Targeting Application EAR and WAR Files for IBM WebSphere Deployment

If you want to deploy WebCenter Portal applications to IBM WebSphere you must ensure that the application's WAR deployment profile and EAR deployment profile are configured with **Platform** set to **WebSphere** (Figure 5–49). For details, see "Creating a WAR Deployment Profile" and "Creating an Application-level EAR Deployment Profile" in *Oracle Fusion Middleware Fusion Developer's Guide for Oracle Application Development Framework*.

Figure 5–49 Configure WebSphere Targeted EAR and WAR Deployment Profiles



5.3.4 Deploying WebCenter Portal Application EARs using WebSphere Console and wsadmin

If your WebCenter Portal application is packaged in an EAR file, you can use the IBM WebSphere Console or wsadmin command (`AdminApp.install`) to deploy the application to WebSphere:

- Deployment Prerequisites
- Deploying WebCenter Portal Application EARs using WebSphere Admin Console
- Deploying WebCenter Portal Application EARs using wsadmin Commands

> **Note:** Applications do not start automatically after deployment or redeployment from JDeveloper. You have to start the WebCenter Portal application manually using the IBM WebSphere Administrative Console or using wsadmin commands.

5.3.4.1 Deployment Prerequisites

Before you deploy the EAR file:

- Ensure that the WAR and EAR deployment descriptors used to generate the EAR file specify "WebSphere" as the target platform. See Section 5.3.3, "Targeting Application EAR and WAR Files for IBM WebSphere Deployment"
- Update the archive's MDS configuration using the wsadmin command `MDSAdmin.getMDSArchiveConfig` and `archive.setAppMetadataRepository`.

  For example:

```
wsadmin>archive =
MDSAdmin.getMDSArchiveConfig(fromLocation='/scratch/oracle/jdeveloper/mywork/my
PortalFwkApp/deploy/myPortalFwkApp_application1.ear')
wsadmin>archive.setAppMetadataRepository(repository='mds-CustomPortalDS',partit
ion='myPortalFwkApp_application1',type='DB',jndi='jdbc/mds/CustomPortalDS')
Operation "setAppMetadataRepository" successful.
wsadmin>archive.save()
```

  See also "Deploying WebCenter Portal: Framework Applications" in *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal*.

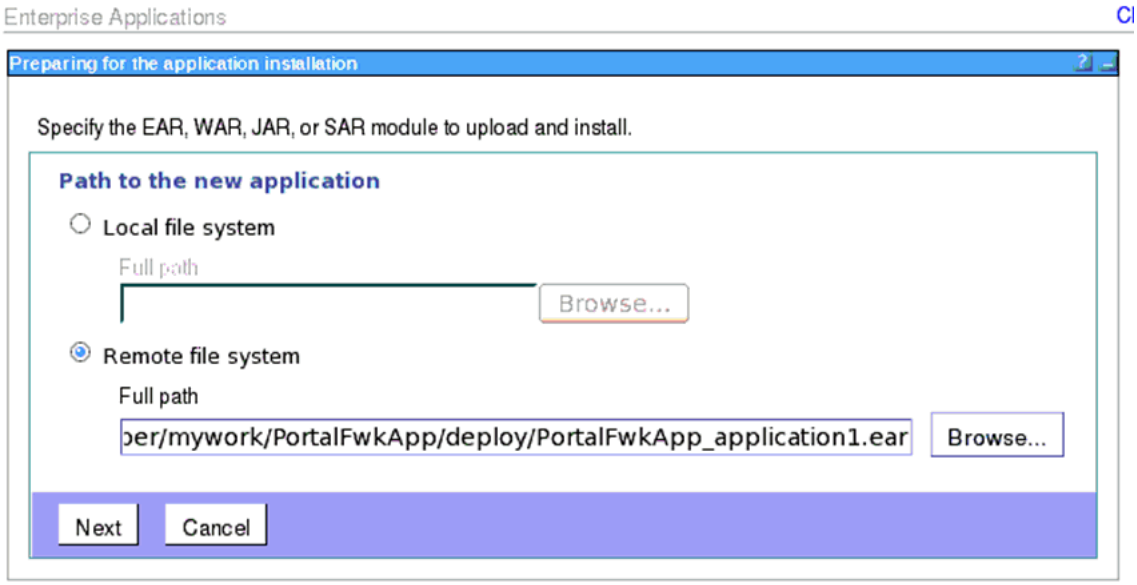
5.3.4.2 Deploying WebCenter Portal Application EARs using WebSphere Admin Console

To deploy a WebCenter Portal application EAR file using IBM WebSphere Console:

> **Note:** For more information, see the IBM WebSphere documentation.

1. Log in to the IBM WebSphere Administrative Console.
2. Navigate to **Applications > New Application >New Enterprise Application**.
3. Enter the location of your application EAR file and click **Next** (Figure 5–50).

Figure 5–50 Specifying the WebCenter Portal Application EAR Location



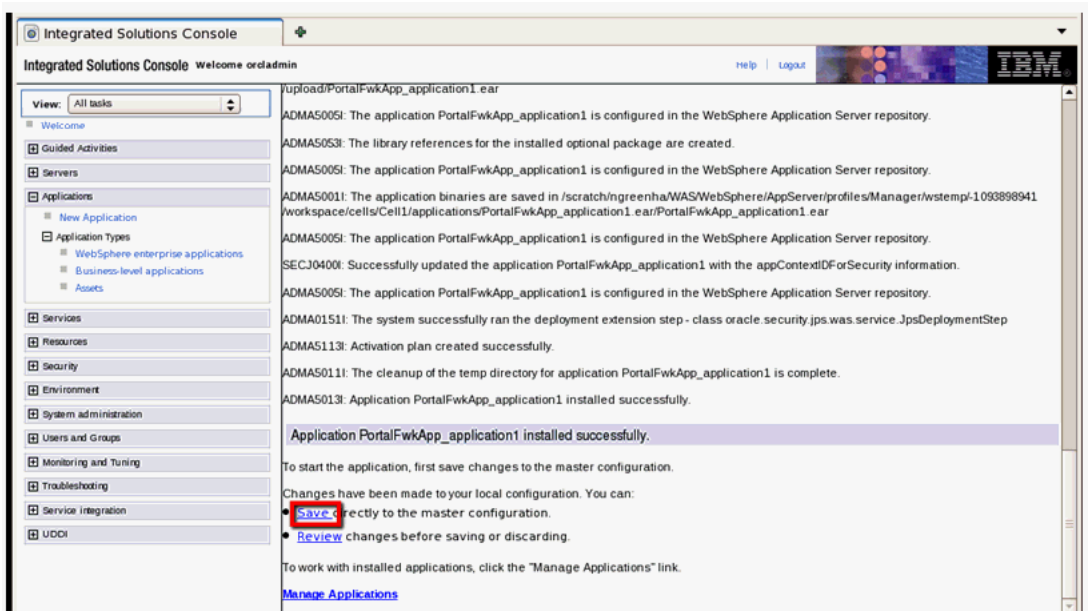
4. On the **Preparing for the application installation page**, accept the default **Fast Path** install option, and click **Next**.

5. On the **Specify options for installing enterprise applications and modules** page, accept all the default settings, and click **Next**.
6. On the **Map modules to servers** page, choose the target server for your application

   For example:

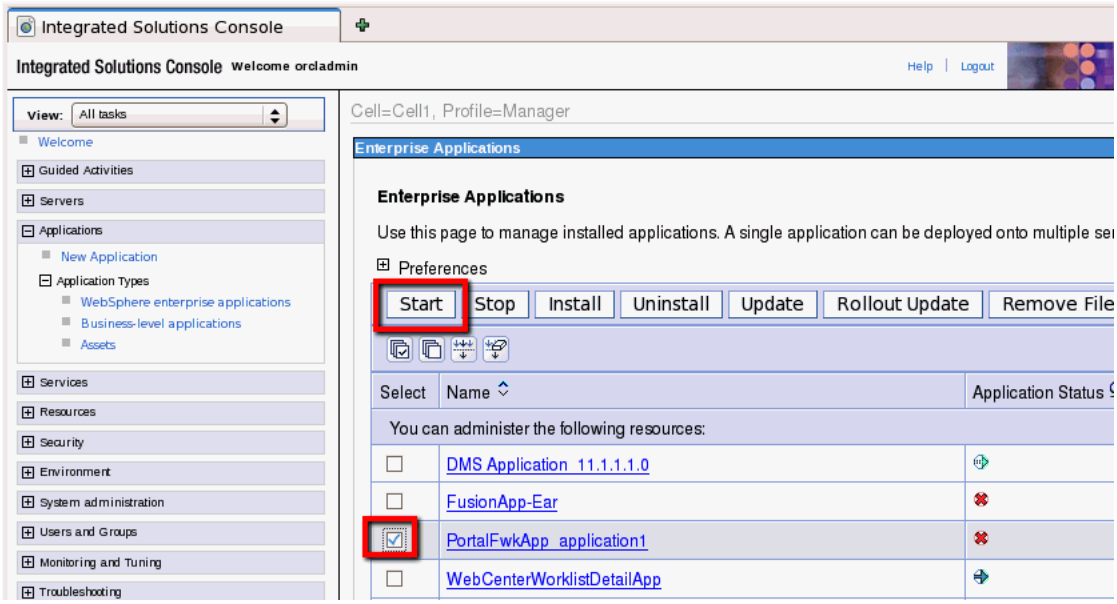
   - For Framework applications, choose **WC_CustomPortal**
   - For Portlet Producer applications, choose **WC_CustomServicesProducer**
7. On the summary page select to **Finish** to install.
8. Select **Save** (Figure 5–51).

Figure 5–51 Saving WebCenter Portal Application EAR Installation



9. Select the name of your newly installed application, and click **Start** (Figure 5–52).

Figure 5–52 Starting the WebCenter Portal Application



Your application is now available.

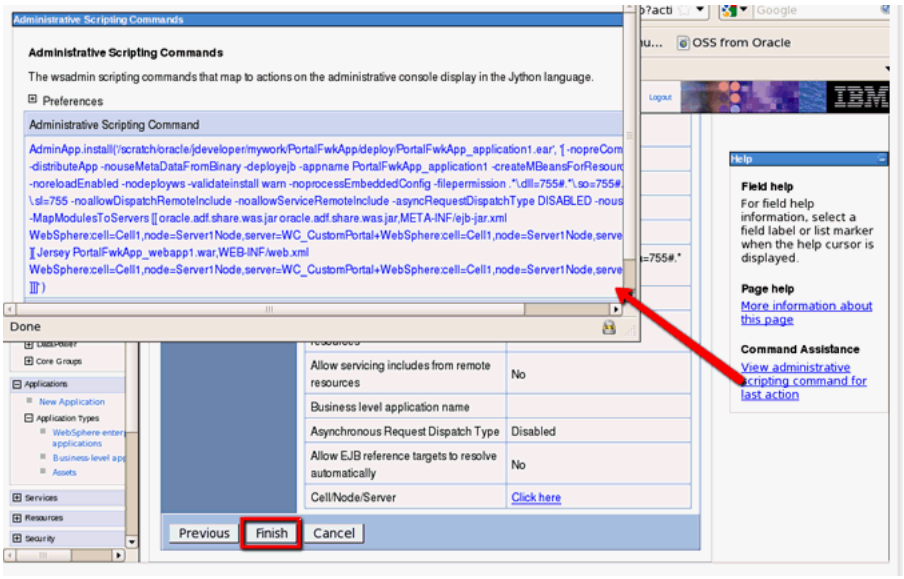
5.3.4.3 Deploying WebCenter Portal Application EARs using wsadmin Commands

The steps in this section recommend using *Command Assistance* to ascertain the correct command syntax and parameter values for application EAR file deployment. While not mandatory, Command Assistance is highly recommended when compiling scripts for lifecycle operations such as deployment.

> **Note:** For more information, see IBM WebSphere documentation.

1. Complete steps 1 through 5 in Section 5.3.4.2, "Deploying WebCenter Portal Application EARs using WebSphere Admin Console".
2. On the summary page, select **View administrative scripting command for last action** (Figure 5–53).

Figure 5–53 Viewing Deployment Scripting Commands



3. Copy the **AdminApp.install** command displayed and paste into a suitable text editor.
4. Edit the EAR file path in the copied command to match the location of your ear file.
5. Deploy your application using the wsadmin command:
   a. Open the wsadmin command prompt connected to the deployment manager.
   b. Paste the updated command.
   c. Execute the wsadmin command.
   d. Save the command.

   For example:

```
wsadmin>AdminApp.install('/scratch/oracle/jdeveloper/mywork/PortalFwkApp/deploy
/PortalFwkApp_application1.ear',
'[ -nopreCompileJSPs -distributeApp -nouseMetaDataFromBinary -deployejb
-appname PortalFwkApp_application1
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall warn
-noprocessEmbeddedConfig -filepermission
.\.dll=755#.\.so=755#.\.a=755#.\.sl=755 -noallowDispatchRemoteInclude
-noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -MapModulesToServers [[
oracle.adf.share.was.jar oracle.adf.share.was.jar,
META-INF/ejb-jar.xml WebSphere:cell=Cell1,node=Server1Node,server=WC_
CustomPortal+WebSphere:cell=Cell1,
node=Server1Node,server=webserver1 ][ Jersey PortalFwkApp_
webapp1.war,WEB-INF/web.xml
WebSphere:cell=Cell1,node=Server1Node,server=WC_
CustomPortal+WebSphere:cell=Cell1,node=Server1Node,server=webserver1 ]]]' )
wsadmin>AdminConfig.save()
```

6. Start the newly deployed WebCenter Portal application using the IBM WebSphere Administrative Console or using wsadmin commands.

   For example:

```
wsadmin>AdminControl.invoke('WebSphere:name=ApplicationManager,process=WC_
CustomPortal,
platform=proxy,node=Server1Node,version=7.0.0.19,type=ApplicationManager,mbeanI
dentifier=ApplicationManager,cell=Cell1,spec=1.0',
'startApplication', '[PortalFwkApp_application1]', '[java.lang.String]')
```

5.3.5 Securing a Framework Application Connection to IMAP and SMTP with SSL

The steps to secure an IMAP/ SMTP connection with SSL for a Framework application deployed on IBM WebSphere are slightly different to that on Oracle WebLogic Server. On WebSphere, you need to specify an additional property in the trust store—`trustStoreType`:

1. Follow the steps "Securing a WebCenter Portal Application's Connection to IMAP and SMTP with SSL" in *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal*.
2. Add the following property to the truststore:

   ```
   -Djavax.net.ssl.trustStore=C:\mail\jssecacerts
   -Djavax.net.ssl.trustStorePassword=changeit
   -Djavax.net.ssl.trustStoreType=JKS
   ```

   For example:

   ```
   set JAVA_PROPERTIES=-Dplatform.home=%WAS_HOME% -Dwls.home=%WAS_HOME%
   -Dweblogic.home=%WAS_HOME%
   -Djavax.net.ssl.trustStore=C:\mail\jssecacerts
   -Djavax.net.ssl.trustStorePassword=changeit
   -Djavax.net.ssl.trustStoreType=JKS
   ```

3. Restart the Framework application.
4. Log into the application and provide your mail credentials.

5.3.6 Using the Deploy and Configure Script for WebCenter Portal Applications Deployed on WebSphere

During its life cycle, a typical portal is deployed to testing, staging, and production servers. WebCenter Portal provides configurable scripts (`create_profile_was.csh` and `deploy_and_config_was.csh`) that allow you to easily deploy and configure Framework applications to these server instances and Oracle recommends that you use these deployment scripts rather than `ojdeploy`.

> **Note:** The deploy and configure scripts in `stage2prod` are samples only. You are free to develop your scripts in a different location (after copying the sample and making changes to it for your deployed environment).

The portal lifecycle and the tasks, tools, and techniques for managing a Framework application deployed on WebLogic Server throughout its life cycle is described in detail in "Understanding the WebCenter Portal Life Cycle" in *Oracle Fusion Middleware Developer's Guide for Oracle WebCenter Portal*. The process is largely the same for WebSphere deployments, however, the script names are different and there is WebSphere-specific content in both `setup.properties` and `profile.properties`.

To deploy and configure an application to a WebSphere using WebCenter Portal scripts:

1. In a terminal window, go to the directory that contains the deploy and configure scripts. These scripts are called: `create_profile_was.csh` and `deploy_and_config_was.csh`. These files reside in `WC_ORACLE_HOME/webcenter/scripts/stage2prod`, where `WC_ORACLE_HOME` is the directory where WebCenter is installed.

   Note: These scripts need access to `wsadmin.properties` and `soap.client.props` files to authenticate and connect to the WebSphere deployment manager's SOAP port. Ensure that both `wsadmin.properties` and `soap.client.props` are present in the directories referenced by the scripts. For more information on how `wsadmin.sh` uses `wsadmin.properties` and `soap.client.props`, refer to IBM WebSphere documentation.

2. Provide target environment-specific information in the `setup.properties` file, such as the target server URL, user name, and password. Open the file `setup.properties` and add the appropriate values for the target environment. A sample file is shown in Example 5–1.

Example 5–1 Sample setup.properties File

```
# WebSphere Server
webcenter.app.node=DefaultCellFederatedNode

# Application
webcenter.app.name=webapp
webcenter.app.server=WC_CustomPortal
webcenter.app.version=V2.0
```

3. Update `create_profile_was.csh` and `deploy_and_config_was.csh` to reflect the deployed environment.

   ```
   setenv WC_HOME <webcenter_home>
   setenv SCRIPTS_DIR <scripts_home>
   ```

   `WC_HOME` is the WebCenter Home and `SCRIPTS_DIR` is where the scripts are located. By default, the scripts are here: `$WC_HOME/webcenter/scripts/stage2prod`. If you copied the scripts to another location, then set `SCRIPTS_DIR` to that location.

4. Run the `create_profile_was` script. The input to this script is the `setup.properites` file. For example, in a Linux environment, enter:

   ```
   ./create_profile_was.csh
   ```

   This script examines your application environment and produces an output properties file called `profile.properties`.

5. If you wish, rename the output file, `profile.properties` to a name that reflects the target environment. For example, if the target environment is your stage environment, you might call the file output file `wcstage.properties`.

   The `profile.properties` file specifies all the configuration information needed to run the portal on the target environment. For example, it includes settings for the content repository, OmniPortlet, WSRP producers, Personalization for

WebCenter Portal and more. Example 5–2 shows a sample `profile.properties` file.

Example 5–2 Sample profile.properties File

```
webcenter.wcps.app.name=wcps-services
webcenter.wcps.app.server=WC_Utilities
doclib.Content.cis.socket.host=hostname
app.mds.jndi=jdbc/mds/SpacesDS
webcenter.app.archive=/net/hostname/scratch/webapp.ear
doclib.Content.cis.socket.port=9444
webcenter.wcps.archive=/net/hostname/scratch/wcps.mar
webcenter.app.name=webapp
app.mds.repository=mds-SpacesDS
app.mds.partition=wcps-services
webcenter.app.version=V2.0
web.OmniPortlet.url=http\://hostname\:7101/portalTools/omniPortlet/providers/omniP
ortlet
app.restart=false
webcenter.app.server=WC_CustomPortal
# Websphere Server SPECIFIC properties
webcenter.app.node=DefaultCellFederatedNode
webcenter.app.deployoptions=[ -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname PortalApp1_application1
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall warn
-noprocessEmbeddedConfig -filepermission
.*\\.dll\=755\#.*\\.so\=755\#.*\\.a\=755\#.*\\.sl\=755
-noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -MapResRefToEJB [[ PortalApp1_
webapp1.war "" PortalApp1_webapp1.war,WEB-INF/web.xml jdbc/WebCenterDS
javax.sql.DataSource jdbc/WebCenterDS "" "" "" ][ PortalApp1_webapp1.war ""
PortalApp1_webapp1.war,WEB-INF/web.xml jdbc/ActivitiesDS javax.sql.DataSource
jdbc/ActivitiesDS "" "" "" ]]
-MapModulesToServers [[ PortalApp1_webapp1.war PortalApp1_
webapp1.war,WEB-INF/web.xml
WebSphere\:cell\=DefaultCell,node\=DefaultCellFederatedNode,server\=WC_
CustomPortal]]
-MapWebModToVH [[ PortalApp1_webapp1.war PortalApp1_webapp1.war,WEB-INF/web.xml
default_host
```

Note: Your environment -specific values will replace the sample values shown in Example 5–2. If a property is not needed, delete it or comment it out rather than leave the value empty.

6. Run `create_profile_was` to create a properties file for each of your target environments. For example, you might create one each for your test, stage, and production environments.
7. Run the `deploy_and_config_was` script. The input to this script is the `profile.properties` file (or whatever you renamed the file). For example, in a Linux environment, might enter:

   ```
   ./deploy_and_config_was.csh wcstage.properties
   ```

The `deploy_and_config_was` script takes one of two "modes" as input. These modes are `deploy_config` and `p13n_metadata`. For example:

```
./deploy_and_config_was.csh p13n_metadata
```

The `deploy_config` mode is the default mode if no input is passed to `deploy_and_config_was.csh`. The `deploy_config` mode does the deployment and configuration tasks. If you only need to update the personalization metadata, you can override the default behavior by passing in `p13n_metadata` as the input to the script.

This script deploys and configures the Framework application to run on the target environment.

5.3.7 Creating SQL Data Controls for Applications Deployed on WebSphere Administration Server

If you want to build SQL data controls for WebCenter Portal applications deployed on IBM WebSphere that use data sources other than the out-of-the-box data sources (WebCenter and Activities), follow the instructions here:

Note: If the SQL data control is consumed in a task flow, the task flow displays only the first 25 rows of data. This is a known limitation.

---

1. Create the custom data sources manually using the IBM WebSphere Administrative Console. See Section 5.3.2.3, "Creating Database Connections to Custom Data Sources".

   However, to be able to create a SQL data control from custom data source, you must configure security alias information as follows:

   Setup security aliases screen:

   – **Component-managed authentication alias** - Select the user connection alias from the Component-managed authentication alias dropdown menu.

   – **Container-managed authentication alias** - Select `none` from the Container-managed authentication alias dropdown menu.

2. Assign administrator roles to users who will create data controls.

   Mbeans are used to access data sources available on the IBM WebSphere Application Server. By default, global security is enabled on the server and only users assigned an administrator role can access Mbeans. Users who are not assigned an administrator role will not be able to see any data sources when they try to create a SQL data control. Therefore, you must assign an administrator role to each user who may need to create SQL data controls.

   To assign an administrator role to a user:

   a. Log in to the IBM WebSphere Administrative Console and navigate to **Security > Global Security**.

   b. Click the **Administrative user roles** link.

   c. Click **Add**.

   d. Select the role (any of deployer, operator, configurator, monitor, administrator, adminsecuritymanager, auditor) and search for the user.

   e. Select the user from the `Available` list and move to the `Mapped to role` list.

   f. Click **OK**.

5.4 Differences Managing WebCenter Portal Components on IBM WebSphere

This section includes the following sub sections

- Section 5.4.1, "Running WebCenter Portal wsadmin Commands"
- Section 5.4.2, "Managing WebCenter Portal Applications With Fusion Middleware Control"

5.4.1 Running WebCenter Portal wsadmin Commands

All WebCenter Portal wsadmin commands have equivalent WLST (WebLogic Scripting Tool) commands which are documented in detail in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

Table 5–6 describes some general differences when running `wsadmin` commands on IBM WebSphere.

Table 5–6 Differences Between WebCenter wsadmin and WLST

Issue	WLST	wsadmin

Command Names	WLST commands are documented in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*. For example:
	`createMailConnection`
	`setWebCenterIdStoreSearchConfig`
	`exportWebCenterApplication`
		All `wsadmin` command names are prefixed with "`WebCenter.`" For example:
		`WebCenter.``createMailConnection`
		`WebCenter.``setWebCenterIdStoreSearchConfig`
		`WebCenter.``exportWebCenterApplication`
		Note: The `WebCenter.` prefix is case sensitive.

Boolean Type	You can use true/false or 1/0.
	For example:
	`setMailConnection(appName='webcenter', name='MyMailServer', default=1)`
	`setMailConnection(appName='webcenter', name='MyMailServer', default=true)`
		You must use 1/0
		For example:
		`WebCenter.setMailConnection (appName='webcenter', name='MyMailServer', default=1)`

applicationVersion	Valid argument.	Not used.
cloneWebCenterManagedServer command	Used to clone WebLogic managed servers when setting up a cluster	Not applicable.

Run WebCenter Portal `wsadmin` commands from the `/common/bin` directory of the Oracle WebCenter Portal home:

```
(UNIX) WC_ORACLE_HOME/common/bin/wsadmin.sh
(Windows) WC_ORACLE_HOME\common\bin\wsadmin.bat
```

To invoke online help for WebCenter Portal commands, enter the following:

```
wsadmin> print OracleHelp.help('WebCenter')
```

To invoke online help for a specific command, enter the command name:

```
wsadmin> print OracleHelp.help('WebCenter.createMailConnection')
```

For more information about `wsadmin` commands, see Section 3.1.3, "Using the Oracle Fusion Middleware wsadmin Commands"."

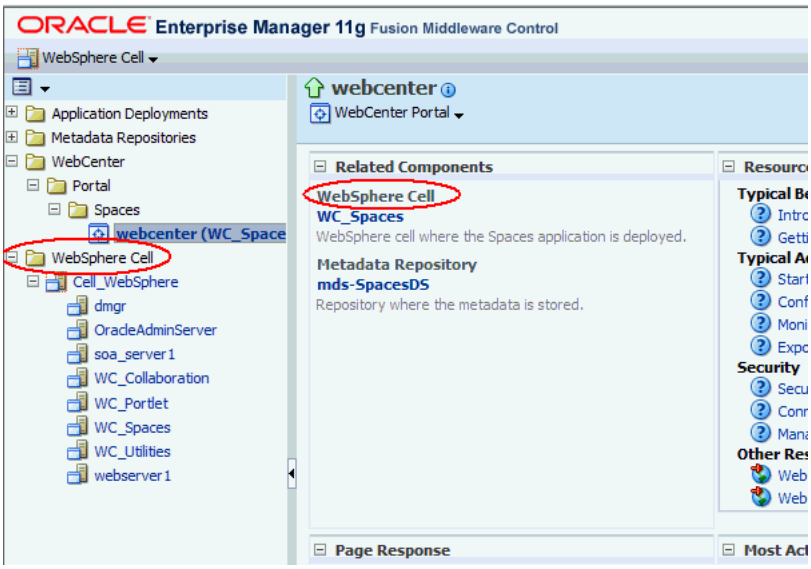
For information about the equivalent WebCenter Portal WLST commands, see *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

5.4.2 Managing WebCenter Portal Applications With Fusion Middleware Control

You can start, stop, or restart a WebCenter Portal cell and manage WebCenter Portal applications deployed on IBM WebSphere with Fusion Middleware Control. The functionality is the same as that described for WebLogic deployments. The only differences are as follows:

- **Navigation Tree** - a WebSphere Cell folder displays in the navigation tree instead of WebLogic Domain folder.
- **Home Page for the Spaces Application** (Figure 5–54)
  - **Related Components section** - you cannot navigate directly to the Spaces application.
  - **Related Components section** - a link to the WebSphere Cell on which the Spaces application is deployed displays instead of a WebLogic Server link.
- **Home Page for Framework Applications** - You cannot navigate to the IBM WebSphere Administrative Console.

Figure 5–54 Fusion Middleware Control Home Page for Spaces Deployment on IBM WebSphere



5.5 Restrictions Using WebCenter Portal on WebSphere

This section describes WebCenter Portal features that are not supported on WebSphere. It contains the following sections:

- Oracle WebCenter Adapter for SharePoint Not Supported on WebSphere
- Process Spaces Not Supported on WebSphere
- Activity Rank for Oracle Secure Enterprise Search Not Supported on WebSphere

5.5.1 Oracle WebCenter Adapter for SharePoint Not Supported on WebSphere

You cannot connect WebCenter Portal application deployments on IBM WebSphere to Microsoft Sharepoint repositories.

5.5.2 Process Spaces Not Supported on WebSphere

The Oracle BPM Process Spaces workspace is not supported on IBM WebSphere for this release (11.1.1.6.0).

5.5.3 Activity Rank for Oracle Secure Enterprise Search Not Supported on WebSphere

The use of Activity Graph ranking to improve the relevancy of Oracle Secure Enterprise Search results is unavailable on IBM WebSphere deployments.

5.6 Troubleshooting WebCenter Portal on WebSphere

Use the information in this section to help troubleshoot issues with WebCenter Portal on WebSphere. It contains the following sections:

- Diagnosing java.lang.RuntimeException or java.lang.NullPointerException
- Connection Timeout Errors
- Session Timeouts in Spaces Applications
- Session Timeouts Due to Inactivity
- Users Can Log In With Old Passwords
- WASX7015E: NameError Exception Running WSADMIN Commands
- Unable to Deploy Spaces Workflows when the SOA MDS schema is Running on DB2

5.6.1 Diagnosing java.lang.RuntimeException or java.lang.NullPointerException

If you attempt to access a WebCenter Portal application that is not yet connected to an identity store, one of the following error messages display:

```
Caused by: java.lang.RuntimeException: User Principal could not be found for
authenticated user.
        at oracle.webcenter.framework.service.Utility.getUserName

Caused by: java.lang.NullPointerException
        at
oracle.webcenter.framework.service.Utility$1.run(Utility.java:1023)
        at
oracle.webcenter.framework.service.Utility$1.run(Utility.java:1020)
at java.security.AccessController.doPrivileged(AccessController.java:251
```

You must install and configure an LDAP ID store for your application. For more information, see Section 5.2.6, "Installing External LDAP ID Store for WebCenter Portal Applications".

5.6.2 Connection Timeout Errors

If your application is processing large data sets you might experience timeout errors. To prevent frequent timeouts, consider increasing the `requestTimeout` property for wsadmin commands and for Enterprise Manager.

The default timeout values are as follows:

- For the call from the wsadmin environment to the deployment manager. The default for is 180 seconds.
- For the connection between the deployment manager and the node agent, the default is 600 seconds.
- For the connection between the node agent and the runtime deployment target, the default is 600 seconds.

To modify the com.ibm.SOAP.requestTimeout property for wsadmin commands:

1. Edit the Deployment Manager `soap.client.props` file.
2. Modify the `com.ibm.SOAP.requestTimeout` value. Enter a value in seconds.

   For example, enter 18000 for a 5 hour timeout.
3. Restart Deployment Manager.
4. Restart the OracleAdminServer.

To modify the Request Timeout for Enterprise Manager:

1. Log in to the IBM Administrative Console.
2. Navigate to: **Servers> Server Types> WebSphere application servers> OracleAdminServer> Container Settings> Container Services> ORB service**
3. Update the value for "Request timeout"
4. Restart the OracleAdminServer.

5.6.3 Session Timeouts in Spaces Applications

Two different settings drive the session timeout in a Spaces application:

- Global timeout (**LTPA timeout** property)
- Application timeout (**wcSessionTimeoutPeriod** attribute in `webcenter-config.xml`)

The lowest of these two values determine the session timeout that is used.

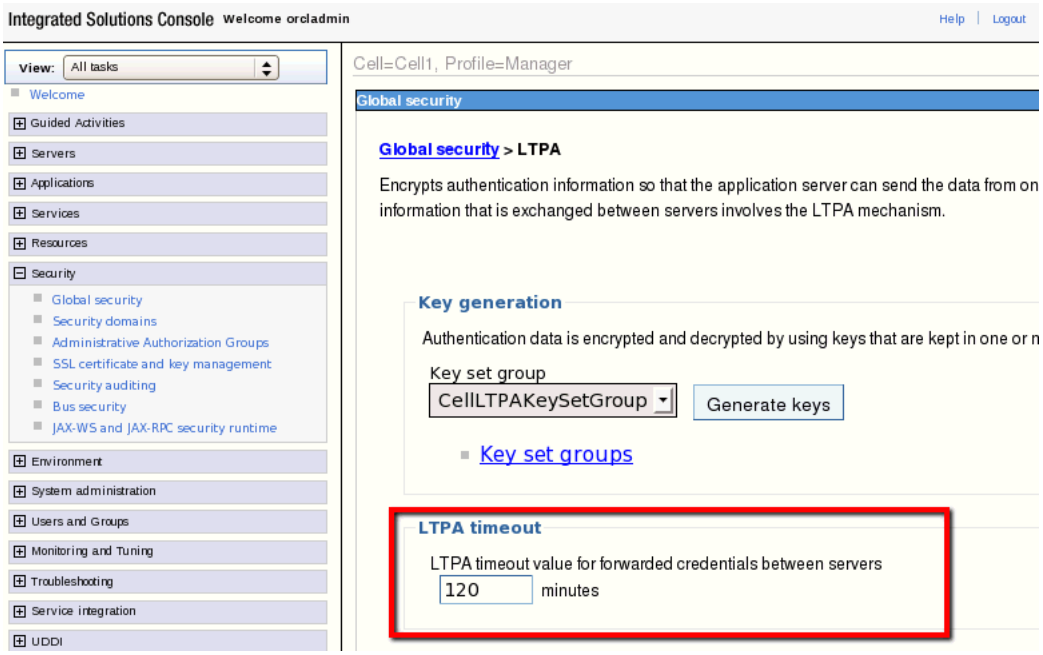
Oracle recommends that you set the global LTPA timeout to be a minute longer than the setting in `webcenter-config.xml` so that users automatically navigate to the Spaces session timeout page.

To set the LTPA timeout:

1. Determine the current value of **wcSessionTimeoutPeriod**.

   To find out how to export the latest `webcenter-config.xml` from MDS, see "Setting a Session Timeout for the Spaces Application" in *Oracle Fusion Middleware Administrator's Guide for Oracle WebCenter Portal*.
2. Log in to the IBM Administrative Console.
3. Navigate to: **Security> Global Security> LTPA**
4. Set **LTPA timeout** (Figure 5–55).

Figure 5–55 LTPA Timeout



5. Restart the servers.

5.6.4 Session Timeouts Due to Inactivity

By default, application modules deployed on IBM WebSphere have their cookie path set to "/". If one or more WAR modules running on the same server as your WebCenter Portal application's WAR have the same cookie path, you may encounter the following message:

```
Because of inactivity, your session has timed out and is no longer active. Click
OK to reload page
```

If you encounter such messages, specify a unique cookie path for each application WAR.

For example, if your Spaces application is using space membership workflows and these workflows are deployed on a SOA server that is in the same cell as WebCenter Portal: Spaces, you must update the cookie path for the **JSessionID** cookie to match the module name. In this case the module name is WebCenterWorklistDetail, so set the "**Cookie Path**" property to **/WebCenterWorklistDetail**.

For detailed steps, see Section 5.2.11, "Setting Cookie Paths for WebCenter Portal Application Modules Post Deployment".

5.6.5 Users Can Log In With Old Passwords

User credentials are cached by default on IBM WebSphere. If you change your password, the old password may still work until you enter your new password. Credential caching is controlled through the security cache property: `com.ibm.websphere.security.util.authCacheEnabled`

If you want to turn off user credential caching, update the JVM setting as follows:

```
com.ibm.websphere.security.util.authCacheEnabled=false
```

Note: Setting this property to `false` impacts performance so Oracle recommends the default setting (`true`).

See also, IBM WebSphere documentation at:
`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tsec_authusers.html`

5.6.6 WASX7015E: NameError Exception Running WSADMIN Commands

If you run WSADMIN commands without the required prefix (`WebCenter.`) or enter an incorrect prefix you see a `NameError` similar to that below:

```
wsadmin>webcenter.listWorklistConnections('webcenter')
WASX7015E: Exception running command:
"webcenter.listWorklistConnections('webcenter')"; exception information:
com.ibm.bsf.BSFException: exception from Jython:
Traceback (innermost last):
File "<input>", line 1, in ?
NameError: webcenter
```

In this example, the incorrect prefix `webcenter.` must be replaced with `WebCenter.`, that is:

```
wsadmin>WebcCenter.listWorklistConnections('webcenter')
```

See also Section 5.4.1, "Running WebCenter Portal wsadmin Commands".

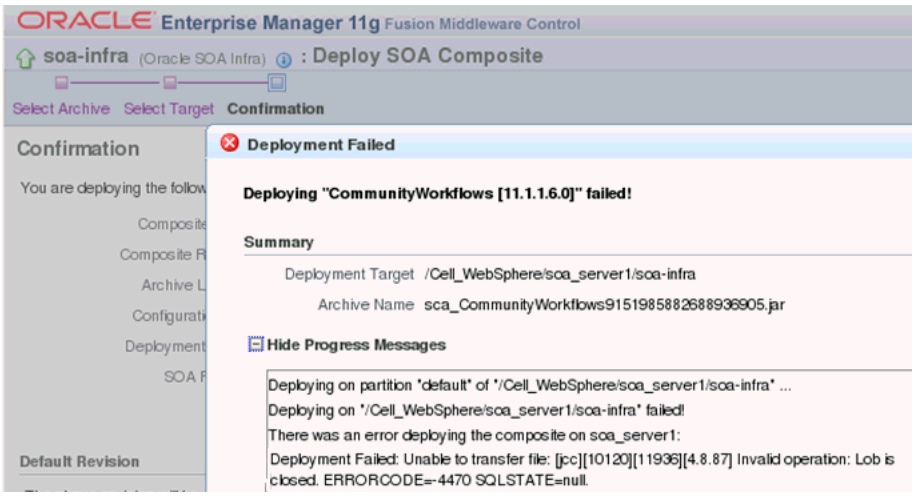
5.6.7 Unable to Deploy Spaces Workflows when the SOA MDS schema is Running on DB2

The composite that manages Spaces workflows (**sca_CommunityWorkflows**) sometimes fails to deploy on a SOA server whose MDS schema is running on a DB2 database.

In such cases, the following message displays in Enterprise Manager (Figure 5–56):

```
Deploying on partition "default" of "/Cell_WebSphere/soa_server1/soa-infra" ...
Deploying on "/Cell_WebSphere/soa_server1/soa-infra" failed!
There was an error deploying the composite on soa_server1: Deployment Failed:
Unable to transfer file: [jcc][10120][11936][4.8.86] Invalid operation: Lob
is closed. ERRORCODE=-4470 SQLSTATE=null.
```

Figure 5–56 scs_CommunityWorkflows Fails to Deploy



If you want to use the Spaces workflows to manage space membership or want to deploy any other BPEL composite on a DB2 back end, modify JBDC data source properties as follows:

1. Log in to the IBM Administrative Console.
2. Navigate to: **Resources> JDBC> Data Sources> mds-soa**
3. Select **Custom properties**.
4. Click **New** and create a custom property with the following values

   Name: progressiveStreaming

   Value: 2

   Description: Disable Progressive Streaming to read lob after result set is closed.
5. Click **OK**.
6. Restart the SOA server.
7. In Enterprise Manager, confirm that the required BPEL composite is now deployed and available as expected.

6

Managing Oracle WebCenter Content on IBM WebSphere

This chapter contains information about managing Oracle WebCenter Content applications on IBM WebSphere Application Servers. It describes differences in performing some Oracle WebCenter Content installation, configuration, and administration tasks on IBM WebSphere from performing these tasks on Oracle WebLogic Server. It also specifies restrictions for some tasks on IBM WebSphere.

This chapter contains the following sections:

6.1 Installing Oracle WebCenter Content on IBM WebSphere

The following sections describe differences for performing some Oracle WebCenter Content installation tasks on IBM WebSphere instead of on Oracle WebLogic Server:

6.1.1 Changing Java Socket Factories in the IBM JDK

If you are using the IBM JDK with Oracle WebCenter Content and IBM WebSphere, certain functionality will not work correctly unless the Java socket factories are changed. For example, during installation, the check for patches feature would fail to connect to Oracle Support. The IBM JRE has its own Secure Sockets Layer (SSL) socket factories, which you need to change to the default JSSE implementation before installing Oracle WebCenter Content.

To change Java socket factories in the IBM JDK:

1. Open the *WAS_HOME*`/java/lib/security/java.security` file in a text editor.

> **Note:** If you are providing the path to a IBM WebSphere on a Windows operating system, and a directory name in the path includes a space, you need to supply a shortened name, with a tilde character (~) followed by a `1` instead of the character before the space. For example, the default location of a WebSphere Application Server on a Windows operation system is in a subdirectory of Program Files, a directory name that includes a space:
>
> ```
> C:\Program Files\IBM\WebSphere\Appserver
> ```
>
> This location needs to be specified as follows:
>
> ```
> C:\Progra~1\IBM\WebSphere\Appserver
> ```
>
> If you are browsing to this location, the **Browse** button incorrectly populates the field with the space rather than `C:\Progra~1`.

2. Uncomment the default JSSE implementation:

```
# Default JSSE socket factories
ssl.SocketFactory.provider=com.ibm.jsse2.SSLSocketFactoryImpl
ssl.ServerSocketFactory.provider=com.ibm.jsse2.SSLServerSocketFactoryImpl
```

3. Comment out the WebSphere SSL implementation:

```
WebSphere socket factories (in cryptosf.jar)
#ssl.SocketFactory.provider=com.ibm.websphere.ssl.protocol.SSLSocketFactory
#ssl.ServerSocketFactory.provider=com.ibm.websphere.ssl.protocol.SSLServerSocke
tFactory
```

4. Save the file.

Alternatively, you can set these properties before a call is made to the Oracle Universal Installer API.

6.1.2 Installing Oracle WebCenter Content Products on IBM WebSphere

Use the Oracle Fusion Middleware 11*g* WebCenter Content Installer to install the binaries for the following Oracle WebCenter Content products on IBM WebSphere.

- Oracle WebCenter Content (WebCenter Content, which includes Oracle WebCenter Content Server)
- Oracle WebCenter Content: Inbound Refinery (Inbound Refinery)
- Oracle WebCenter Content: Records (Records)

For information about the general installation process on IBM WebSphere, see Chapter 2, "Installing and Configuring Oracle Fusion Middleware on IBM WebSphere." For details about differences that apply to all Oracle Fusion Middleware components, see Section 2.5.2, "Special Instructions When Installing Oracle Fusion Middleware with IBM WebSphere."

> **Note:** The Middleware home directory must be outside of the IBM WebSphere home directory (*WAS_HOME*), on the same host, so that updates to the application server do not affect the Middleware home directory.

The installation instructions are similar to those provided for Oracle WebLogic Server in the "Installing Oracle WebCenter Content" chapter and "Installation Screens" appendix of the *Oracle WebCenter Content Installation Guide*.

6.1.3 Setting JDBC Driver Environment Variables for a DB2 Database

If you are using a DB2 database, you must set the following environment variables to include the full paths to `db2jcc4.jar`, `db2jcc_license_cu.jar`, and `db2jcc_license_cisuz.jar`:

- DB2_JCC_DRIVER_NATIVEPATH
- DB2_JCC_DRIVER_PATH

Do this immediately after installing Oracle WebCenter Content products. If you do not do this, all DB2 connection tests will fail.

To set JDBC driver environment variables for a DB2 Database:

1. Open the IBM WebSphere Administrative Console, at this URL:

   `https://host:port/ibm/console`

2. Log in as an administrator, expand **Environment** on the left of the console, and click **WebSphere variables**.
3. From the **Scope** list on the WebSphere Variables page, choose the node that contains your Oracle WebCenter Content installation.
4. Locate and set the following JBDC variables:
   - DB2_JCC_DRIVER_NATIVEPATH
   - DB2_JCC_DRIVER_PATH

   Specify the location of the required DB2 drivers (`db2jcc4.jar`, `db2jcc_license_cu.jar`, and `db2jcc_license_cisuz.jar`). For example:

   ```
   DB2_JCC_DRIVER_NATIVEPATH = WAS_
   HOME/deploytool/itp/plugins/com.ibm.datatools.db2_2.1.102.v20100709_0407/driver

   DB2_JCC_DRIVER_PATH    WAS_HOME/deploytool/itp/plugins/com.ibm.datatools.db2_
   2.1.102.v20100709_0407/driver
   ```

   In the example, *WAS_HOME* refers to the location where IBM WebSphere is installed, as described in Section 2.4.2.3, "About the WAS_HOME Directory Path."

Note: If you are providing the path to a WebSphere Application Server on a Windows operating system, and a directory name in the path includes a space, you need to supply a shortened name, with a tilde character (~) followed by a `1` instead of the character before the space. For example, the default location of a WebSphere Application Server on a Windows operation system is in a subdirectory of Program Files, a directory name that includes a space:

```
C:\Program Files\IBM\WebSphere\Appserver
```

This location needs to be specified as follows:

```
C:\Progra~1\IBM\WebSphere\Appserver
```

If you are browsing to this location, the **Browse** button incorrectly populates the field with the space rather than `C:\Progra~1`.

5. Save both settings.
6. If you are using a cluster, repeat these steps for each node in the cluster.
7. To test the DB2 connection:
   a. Expand **Resources** and **JDBC** on the left of the console, and click **Data sources**.
   b. Select a data source in the table, and click the **Test Connection** button.

6.2 Configuring Oracle WebCenter Content on IBM WebSphere

The following sections describe differences for performing some configuration tasks for configuring Oracle WebCenter Content on IBM WebSphere instead of on Oracle WebLogic Server:

- Configuring Oracle WebCenter Content on IBM WebSphere
- Specifying Deployment with SSL
- Configuring an LDAP Server for Oracle WebCenter Content Users and Groups on IBM WebSphere
- Configuring an Administration User for WebCenter Content
- Setting Up Node Manager
- Launching the IBM WebSphere Administrative Console
- Increasing the Java VM Heap Size for an Oracle WebCenter Content Application Server
- Configuring the Report Library for Records Management in Content Server
- Configuring Session Persistence in a Clustered Environment
- Using Oracle WebCenter Content wsadmin Commands Instead of WLST Commands

6.2.1 Configuring Oracle WebCenter Content on IBM WebSphere

Configuration of Oracle WebCenter Content on IBM WebSphere is largely the same as the configuration of Oracle WebCenter Content described in the *Oracle WebCenter*

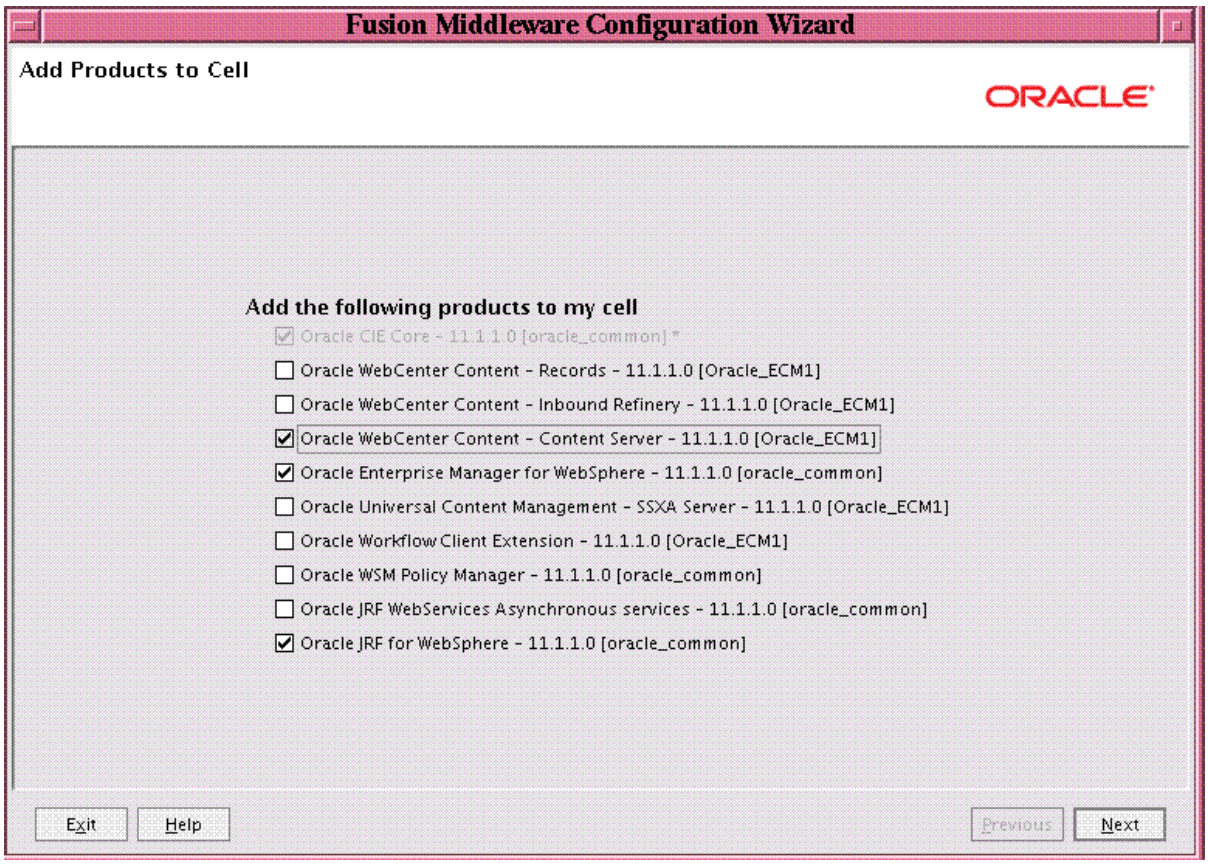
Content Installation Guide. After you have successfully run the Oracle Fusion Middleware 11*g* WebCenter Content Installer and created application schemas, you can deploy and configure WebCenter Content (and Content Server), Inbound Refinery, and Records as applications.

Each Oracle WebCenter Content application is deployed to a WebSphere Application Server Java EE container. The XML files are the same as for Oracle WebLogic Server.

For information about using the Fusion Middleware Configuration Wizard, including information about adding servers and clusters to a cell, see the *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

The configuration screen in Figure 6–1 shows templates for Oracle WebCenter Content applications and related templates that you can add to a WebSphere Application Server cell.

Figure 6–1 Add Products to Cell Screen

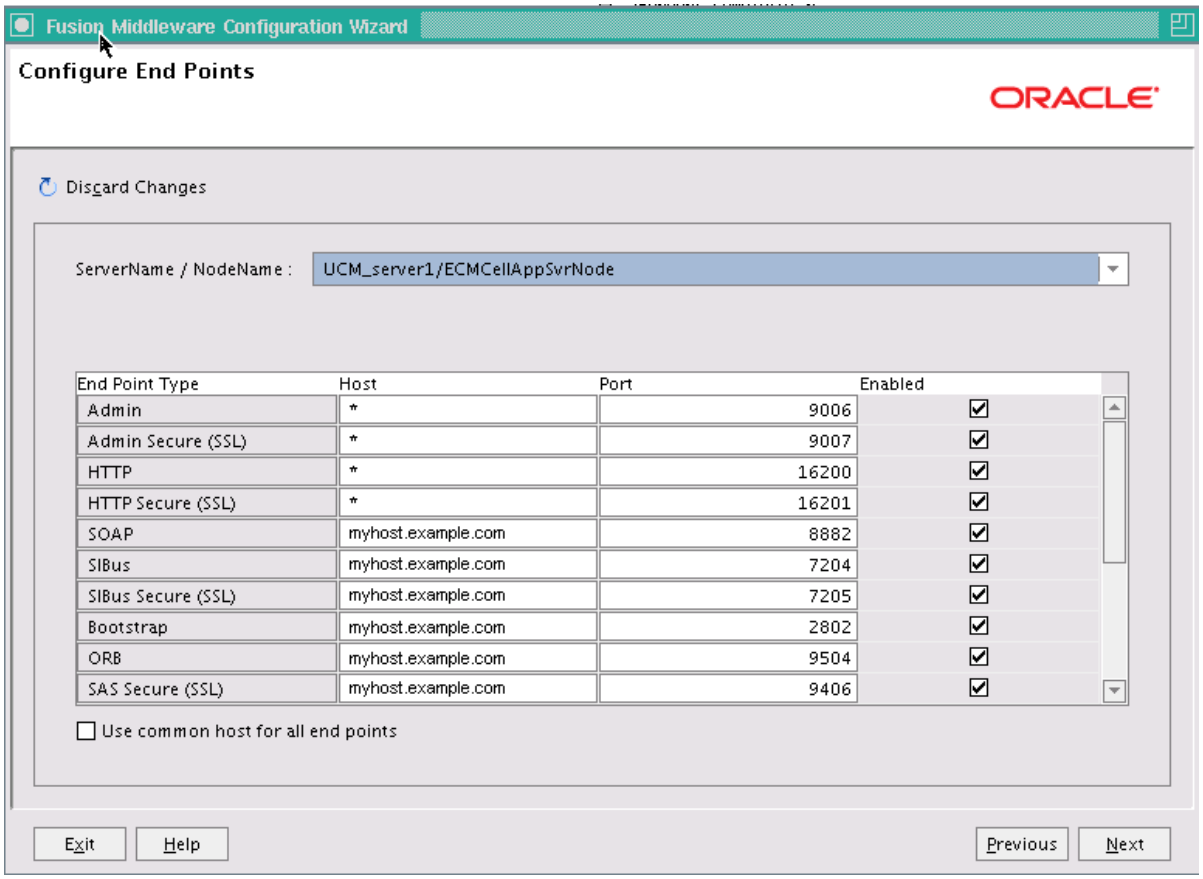


6.2.2 Specifying Deployment with SSL

You can configure WebCenter Content, Inbound Refinery, or Records for deployment with or without Secure Socket Layer (SSL). For this configuration on Oracle WebLogic Server, the Fusion Middleware Configuration Wizard provides an **SSL Enabled** checkbox and **SSL listen port** field for each server on the Configure Managed Servers screen. For SSL configuration on IBM WebSphere, the Configuration Wizard provides an **Enabled** checkbox and **Port** field for each end point on the Configure End Points screen.

All ports, including SSL ports, are enable by default on IBM WebSphere. The **End Point Type** column on the Configure End Points screen indicates whether a port is for SSL, with "`(SSL)`" at the end of the column value. Figure 6–2 shows an example of this screen.

Figure 6–2 Configure End Points Screen



6.2.3 Configuring an LDAP Server for Oracle WebCenter Content Users and Groups on IBM WebSphere

When you configure Oracle WebCenter Content on IBM WebSphere, an internal Lightweight Directory Application Protocol (LDAP) server is *not* automatically configured with users and groups for the WebCenter Content (Content Server), Inbound Refinery, or Records applications. You must manually perform these configuration tasks in an external LDAP server, such as Oracle Internet Directory, after installation and before you start the application servers.

For information about the LDAP servers that Oracle Fusion Middleware supports, see the certification information on the Oracle Technology Network:

`http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html`

For information about installing and configuring a supported LDAP server, see Section 4.1.2, "Configuring Oracle SOA Suite and Oracle BAM in an External LDAP Server." To switch LDAP authentication providers, follow the instructions in this section.

6.2.4 Configuring an Administration User for WebCenter Content

The WebCenter Content administrator has to have an entry in the LDAP authentication provider and in the Administrators group. You can add a user to the Administrators group through Fusion Middleware Control.

6.2.5 Setting Up Node Manager

After using the Fusion Middleware Configuration Wizard to install and configure Oracle WebCenter Content products on IBM WebSphere, start the IBM WebSphere Deployment Manager, Node Manager, and application servers, as described in Section 2.8, "Task 8: Start the IBM WebSphere Servers."

For Oracle WebCenter Content, you must also run the `syncNode` script, as follows:

```
WAS_HOME/bin/syncNode.sh localhost SOAP_CONNECTOR_ADDRESS -profileName profile_
name -username was_admin_user -password was_password
```

The *SOAP_CONNECTOR_ADDRESS* is the value of `Management SOAP connector port` in the *WAS_HOME*`/profiles/`*profile_name*`/logs/AboutThisProfile.txt` or `com.ibm.ws.scripting.port` in the *WAS_HOME*`/profiles/`*profile_name*`/properties/wsadmin.properties` file.

> **Note:** The `syncNode` script should be run only one time, during the first startup sequence. Run it after step 1, "Start the Deployment Manager," in Section 2.8, "Task 8: Start the IBM WebSphere Servers."

6.2.6 Launching the IBM WebSphere Administrative Console

The IBM WebSphere Administrative Console provides a web-based interface for managing the WebSphere Application Server environment. The IBM WebSphere Administrative Console is similar to the Oracle WebLogic Server Administration Console. You cannot use the IBM WebSphere Administrative Console to manage the Oracle WebCenter Content applications, but you can use the console to monitor and manage the cell and the servers on which WebCenter Content, Inbound Refinery, and Records are deployed. For more information, see Section 3.1.1, "Using the WebSphere Administrative Console."

6.2.7 Increasing the Java VM Heap Size for an Oracle WebCenter Content Application Server

You need to increase the size of the heap allocated for the Java Virtual Machine (VM) on which each Oracle WebCenter Content application runs to at least 1 GB (1024 MB) for the IBM JDK. If you do not increase the Java VM heap size, then Oracle support and development will not accept escalation of runtime issues, especially out-of-memory issues.

You can use the IBM WebSphere Administrative Console to adjust the heap size for a Java VM. To increase the heap size, you set the values of the JVM startup parameter.

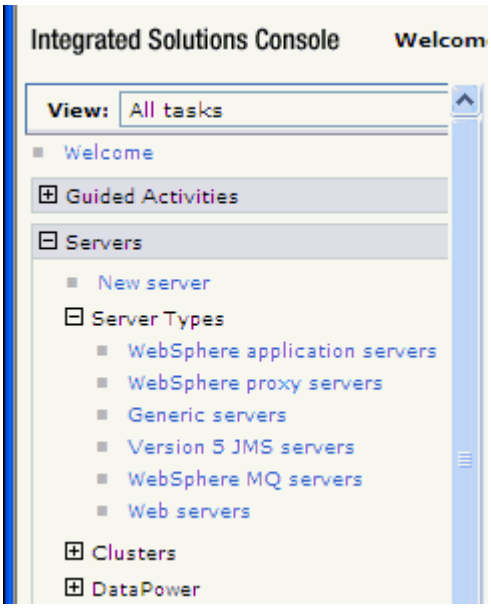
To increase the Java VM heap size for an Oracle WebCenter Content application server with the IBM WebSphere Administrative Console:

1. Log in to the IBM WebSphere Administrative Console as an administrator at `https://`*hostname*`:`*WC_Adminhost_port*`/ibm/console`; for example:

   ```
   https://host42.example.com:9002/ibm/console
   ```

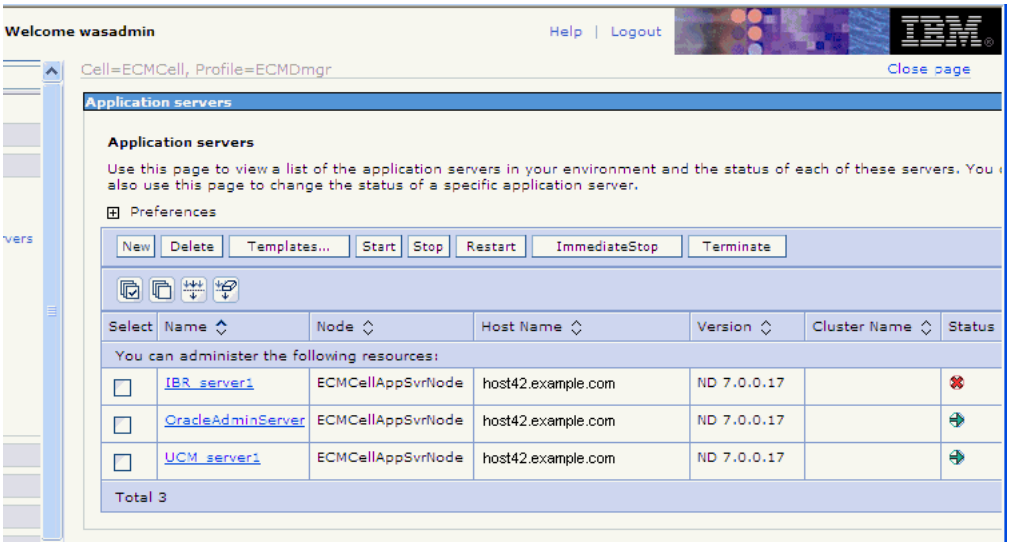
2. In the navigation tree on the left, shown in Figure 6–3, expand **Servers** and **Server Types**, and click **WebSphere application servers**.

Figure 6–3 Server Types



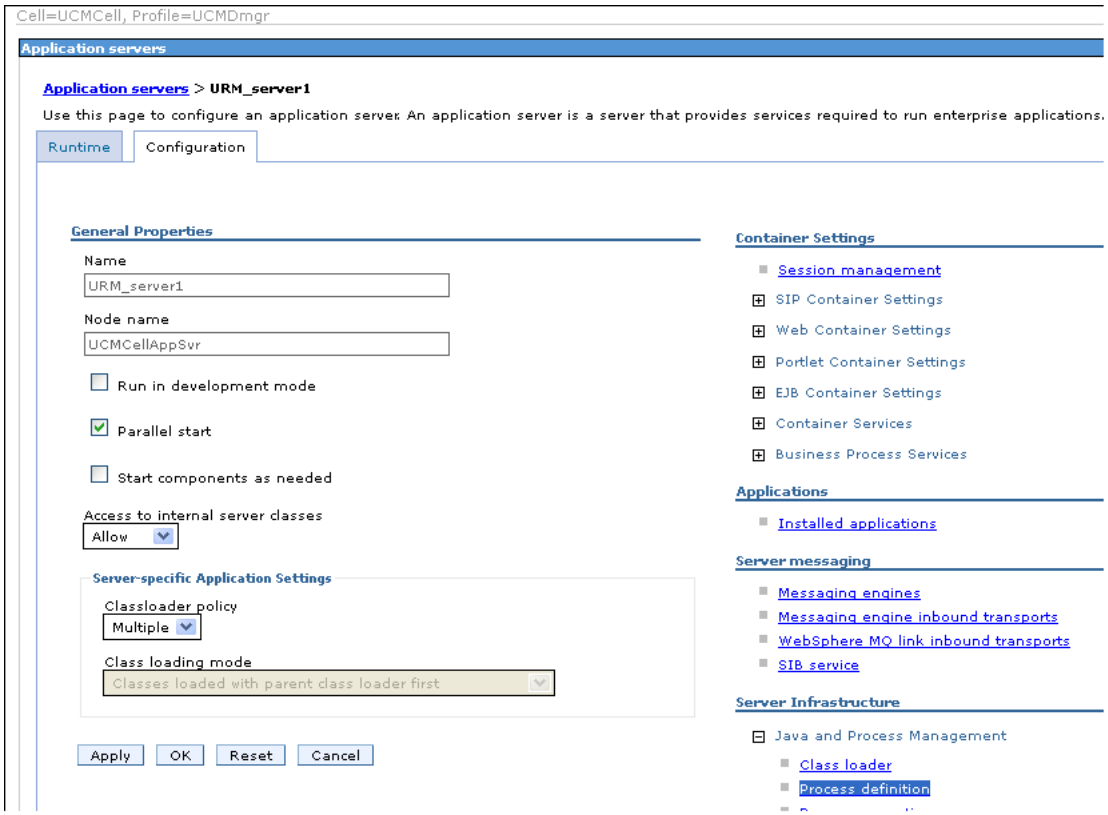
3. On the Application servers page, shown in Figure 6–4, click an Oracle WebCenter Content application server for which you want to increase the heap size.

Figure 6–4 Application servers Page



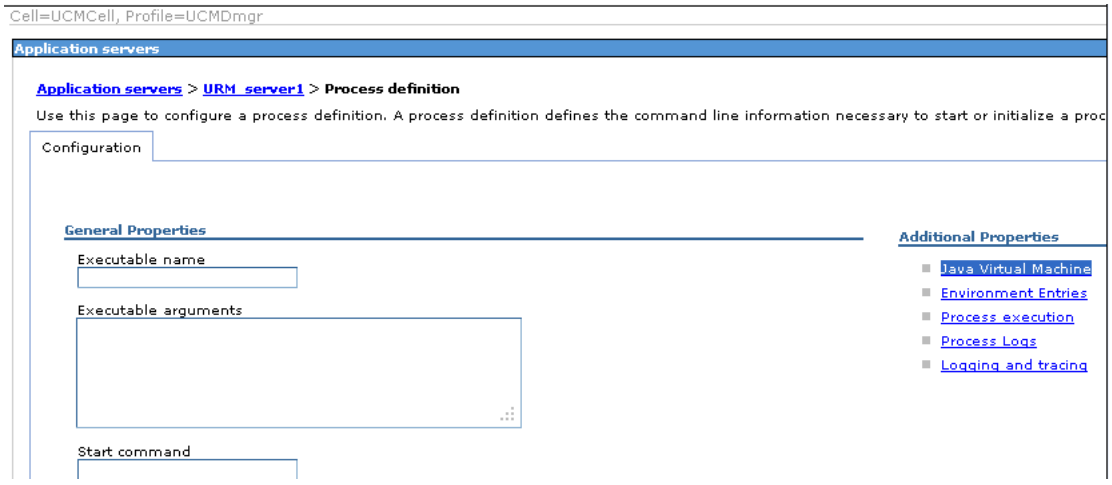
4. On the General Properties page, shown in Figure 6–5, expand **Java Process Management**, and click **Process Definition**.

Figure 6–5 General Properties Page



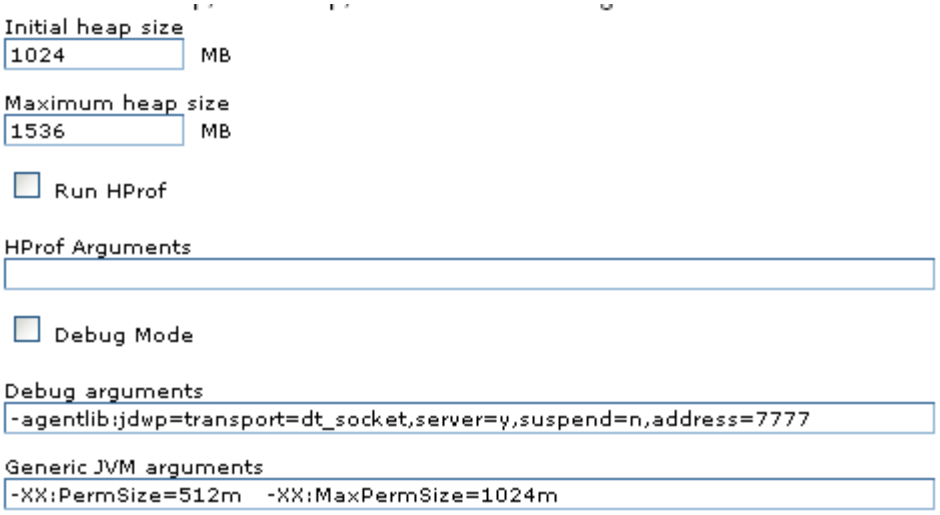
5. On the Process Definition page, shown in Figure 6–6, click **Java Virtual Machine** under Additional Properties.

Figure 6–6 Process definition Page



6. Set the **Initial Heap**, **Max Heap**, and **Generic JVM Arguments** fields to the values shown in Figure 6–7.

Figure 6–7 JVM Startup Parameters



The **PermSize** value in the **Generic JVM Arguments** section needs to be half of the **MaxPermSize** value.

7. Save the changed values to the local configuration.
8. Repeat steps 3 through 7 for each Oracle WebCenter Content application server.
9. Restart the Oracle WebCenter Content application servers, as described in Section 3.2.2, "Starting and Stopping Servers on IBM WebSphere."

6.2.8 Configuring the Report Library for Records Management in Content Server

If you plan to configure the Records Management feature in Content Server, you need configure the report library for Records Management after the server node is created for WebCenter Content, before starting the server for the first time. Without this library, you cannot check in any templates to Content Server.

To configure the report library, you need to add the `oracle.xdo.runtime.ear` library manually from the IBM WebSphere Administrative Console.

Note: This library is not needed for Oracle WebCenter Content: Records

---

To configure the report library for Records Management in Content Server:

1. From the IBM WebSphere Administrative Console, click **Shared Libraries** under **Environment**, on the left.
2. In the shared libraries display, select the scope value as the WebCenter Content node that needs the library.
3. If the `oracle.xdo.runtime.ear` library does not already exist in the Shared Libraries list, click **New**, and enter these values:

   ```
   Name=oracle.xdo.runtime_1_11.1.1.3.0
   Classpath=
   WC_CONTENT_ORACLE_HOME/ucm/idc/components/ReportPublisher/lib/APP-INF/lib
   ```

4. Click **Apply**.

5. In the Servers section on the left, under Server Types, click **WebSphere application servers**.
6. In the table on the Application servers screen, click **Oracle WebCenter Content**.
7. In the installed applications display, click **WebCenter Content**.
8. In the application configuration display, under References, click **Shared library references**.
9. Select the **Oracle WebCenter Content** application, and click the **Reference Shared Libraries** button.
10. In the Shared Library Mapping display, select **oracle.xdo.runtime_1_11.1.1.3.0** on the left, move the selection to the right, and click **OK**.
11. In the Reference Shared Libraries display, click **OK**.
12. Under Messages at the top, click **Save** to save the local configuration to the master configuration.

For information about adding the Records Management feature to Content Server, see "Configuring Records Management in Content Server" in the *Oracle WebCenter Content Installation Guide*.

6.2.9 Configuring Session Persistence in a Clustered Environment

The configuration for session persistence in a clustered environment is different for WebSphere Application Server than for Oracle WebLogic Server, in which the element `<persistent-store-type>replicated_if_clustered</persistent-store-type>` can be set in the servlet to provide session persistence. In WebSphere Application Server, the web container's session management property for session persistence is `sessionPersistenceMode`, which is stored in the *`WAS_HOME`*`/profiles/`*`profile_name`*`/config/cells/`*`cell_name`*`/nodes/`*`node_name`*`/servers/`*`server_name`*`/server.xml` file.

You can display the current `sessionPersistenceMode` value in the IBM WebSphere Administrative Console.

You can set the session persistence mode while creating a cluster through the IBM WebSphere Administrative Console. If you use the Fusion Middleware Configuration Wizard to create a cluster, after the cluster is created, you must configure the session persistence mode manually, through the IBM WebSphere Administrative Console, and then restart the application servers.

To display the session persistence mode in the IBM WebSphere Administrative Console:

1. Log in to the IBM WebSphere Administrative Console as an administrator at `https://`*`hostname`*`:`*`WC_Adminhost_port`*`/ibm/console`.
2. Expand **Servers** and **Server Types** on the left, and click **WebSphere application servers**.
3. Click an Oracle WebCenter Content application server on the Application Servers page.
4. Expand **Web Container Settings**, and click **Web container**, under Container Settings on the Configuration page for the application server.
5. Click **Session management**, under Additional Properties on the Web container page.

6. Click **Distributed environment settings**, under Additional Properties on the Session management page.
7. The Configuration page for Distributed environment settings displays the current `sessionPersistenceMode` value under General Properties, as the selected value for Distributed sessions.

   The value of `sessionPersistenceMode` can be `None`, `Database`, or `Data_Replication` (displayed as the **Memory-to-memory replication** mode on the console.

To set the session persistence mode for a cluster:

- If you use the IBM WebSphere Administrative Console to create a cluster, you must select the option **Configure HTTP session memory-to-memory replication**.

  When this option is selected, a replication domain is created automatically, and the `sessionPersistenceMode` property is automatically set to the `Data_Replication` mode for each member of the new cluster. The created `Replication` domain, which has the same name as the cluster name, is automatically set for the **Replication Domain** property.
- If you use the Fusion Middleware Configuration Wizard to create a cluster, the Configuration Wizard does not present any options for cluster creation, and it creates the new cluster without creating a replication domain.

  The `sessionPersistenceMode` property is not set to the `Data_Replication` mode for each cluster member. In this case, you must configure `sessionPersistenceMode` manually, as the following procedure describes.

To configure the session persistence mode manually:

1. Log in to the IBM WebSphere Administrative Console as an administrator at `https://`*`hostname`*`:`*`WC_Adminhost_port`*`/ibm/console`.
2. Create a replication domain (if there is none, or if you want to create a new replication domain for a cluster you created through the Fusion Middleware Configuration Wizard):
   a. Expand **Environment** on the left of the console, and click **Replication domains**.
   b. Click the **New** button on the Replication domains page.
   c. Enter the domain name, and select the option **Entire Domain** under Number of Replicas.
   d. Click the **Apply** or **OK** button, and then **Save**.
3. Set the `sessionPersitenceMode` property value for every server member in the cluster:
   a. Expand **Servers** and **Server Types** on the left, and click **WebSphere application servers**.
   b. Click an Oracle WebCenter Content application server on the Application Servers page.
   c. Expand **Web Container Settings**, and click **Web container**, under Container Settings on the Configuration page for the application server.
   d. Click **Session management**, under Additional Properties on the Web container page.

e. Click **Distributed environment settings**, under Additional Properties on the Session management page.

f. Select the **Memory-to-memory replication** option, under Distributed sessions on the Configuration page for Distributed environment settings (to set `sessionPersistenceMode` to `Data_Replication`.

g. On the Configuration page for Memory-to-memory replication, select the replication domain you created for this cluster.

h. Select the proper Replication mode for your configuration.

i. Click the **Apply** or **OK** button, and then **Save**.

4. Restart the Oracle WebCenter Content application servers, as described in Section 3.2.2, "Starting and Stopping Servers on IBM WebSphere."

6.2.10 Using Oracle WebCenter Content wsadmin Commands Instead of WLST Commands

All Oracle WebCenter Content `wsadmin` commands supported by WebSphere Application Server have equivalent WebLogic Scripting Tool (WLST) commands. Table 6–1 describes differences between `wsadmin` and WLST.

Table 6–1 Differences Between wsadmin and WLST Commands

Issue	WLST	wsadmin
Command Names	Documented in "Oracle WebCenter Content Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*	All `wsadmin` command names are prefixed with "`UCM.`" For example: **`UCM.`**`getUCMHttpServerAddress` **`UCM.`**`setUCMServerPort` **`UCM.`**`getUCMMailServer`
Boolean Type	`true/false` or `1/0`.	`1/0` only
server, applicationVersion	Valid arguments	Not used
Offline or Online	Run WLST commands in offline mode	Run `wsadmin` commands in online mode
Clone command	Used to clone WebLogic managed servers when setting up a cluster	n/a

Note: The `wsadmin` online commands using MBeans may not provide specific error details. Instead, you may see just an `MBeanException` message.

Execute Oracle WebCenter Content `wsadmin` commands from the `/common/bin` directory in the WebCenter Content Oracle home:

```
cd WC_CONTENT_ORACLE_HOME/common/bin
./wsadmin.sh
```

To invoke online help for Oracle WebCenter Content commands, enter the following command:

```
wsadmin> print OracleHelp.help('UCM')
```

To invoke online help for a specific command, enter the following command:

```
wsadmin> print OracleHelp.help('command_name')
```

To use the commands, you must be connected to a running WebSphere Application Server instance that has the UCM Config MBeans deployed. The MBeans are typically installed in the server. To connect to a server instance, run the `wsadmin.sh` script with these options:

```
./wsadmin.sh -conntype SOAP -port port -user username -password password
```

For example:

```
./wsadmin.sh -conntype SOAP -port 8879 -user wasadmin -password password
```

Table 6–2 shows the `wsadmin` commands that are available for Oracle WebCenter Content server configuration.

Table 6–2 wsadmin Commands for Oracle WebCenter Content

wsadmin Command	Description
`UCM.getUCMCSVersion`	Gets the version of the running instance of Content Server.
`UCM.getUCMHttpServerAddress`	Returns the Content Server HTTP Server Address.
`UCM.getUCMIPAddressFilter`	Gets the IP Address Filter Configuration Parameter.
`getUCMMailServer`	Returns the Content Server Mail Server Configuration Value.
`UCM.getUCMServerPort`	Gets the Server Port Configuration Parameter from the `config.cfg` file and displays it.
`UCM.getUCMServerUptime`	Gets the amount of time the Content Server instance has been up.
`UCM.getUCMSmtpPort`	Gets the Content Server SMTP Port Value.
`UCM.getUCMSysAdminAddress`	Gets the Content Server Administrator Mail Address from the `config.cfg` file.
`UCM.getUCMUseSSL`	Gets the SSL Value from the `config.cfg` file and displays the value as `true` or `false`.
`UCM.setUCMHttpServerAddress`	Sets the Content Server HTTP Server Address.
`UCM.setUCMIpAddressFilter`	Sets the IP Address Filter Configuration Parameter.
`UCM.setUCMMailServer`	Sets the Content Server Mail Server Configuration Value.
`UCM.setUCMServerPort`	Sets the Server Port Configuration Parameter.
`UCM.setUCMSmtpPort`	Sets the Content Server SMTP Port Value.
`UCM.setSysAdminAddress`	Sets the Content Server Administrator Mail Address.
`UCM.setUCMUseSSL`	Sets the SSL Value to `true` or `false`, thereby enabling or disabling the use of SSL.

Example 6–1 shows the syntax and an example of the `UCM.getUCMMailServer` command.

Example 6–1 Get the Content Server Mail Server

```
Syntax:  UCM.getUCMMailServer(AppName)

Example: UCM.getUCMMailServer('Oracle WebCenter Content - Content Server')
```

Example 6–2 shows the syntax and an example of the `UCM.setUCMServerPort` command.

Example 6–2 Set the Content Server Port

```
Syntax:  UCM.setUCMServerPort(value,AppName)

Example: UCM.setUCMServerPort(4444,'Oracle WebCenter Content - Content Server')
```

Example 6–3 shows the help output and an example for the `UCM.setUCMServerPort` command.

Example 6–3 Get the Content Server Version

```
wsadmin>print OracleHelp.help('UCM.getUCMCSVersion')


Gets the value of Content Server Version from the Content Server API's and
displays it.



Syntax:

getUCMCSVersion() or getUCMCSVersion(application_name).



Example:

Example Output: 11g.1.1.0
Example Setting: getUCMCSVersion('Oracle WebCenter Content Server')


wsadmin>UCM.getUCMCSVersion()
11gr1-trunk-idcprod1-111007T175404(Build: 7.3.3.183)
wsadmin>
```

For more information about `wsadmin` commands, see Section 3.1.3, "Using the Oracle Fusion Middleware wsadmin Commands."

For information about the equivalent Oracle WebCenter Content WLST commands, see "Oracle WebCenter Content Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

6.3 Configuring Oracle WebCenter Content Applications on IBM WebSphere

For information about the postinstallation configuration of Oracle WebCenter Content applications, see these chapters in the *Oracle WebCenter Content Installation Guide*:

- "Configuring Oracle WebCenter Content Applications"
- "Completing the WebCenter Content Configuration"

- "Completing the Inbound Refinery Configuration"
- "Completing the Records Configuration"

The following sections describe differences for performing some postinstallation configuration tasks for Oracle WebCenter Content applications on IBM WebSphere instead of on Oracle WebLogic Server:

- Mapping the weblayout Directory
- Changing the Authentication Method for Oracle WebCenter Content Applications

---

Important: The first user to log in to Oracle WebCenter Content Server must be the administrator of the cell, to complete the configuration of Content Server. For more information, see "Completing the Initial WebCenter Content Configuration" in the *Oracle WebCenter Content Installation Guide*.

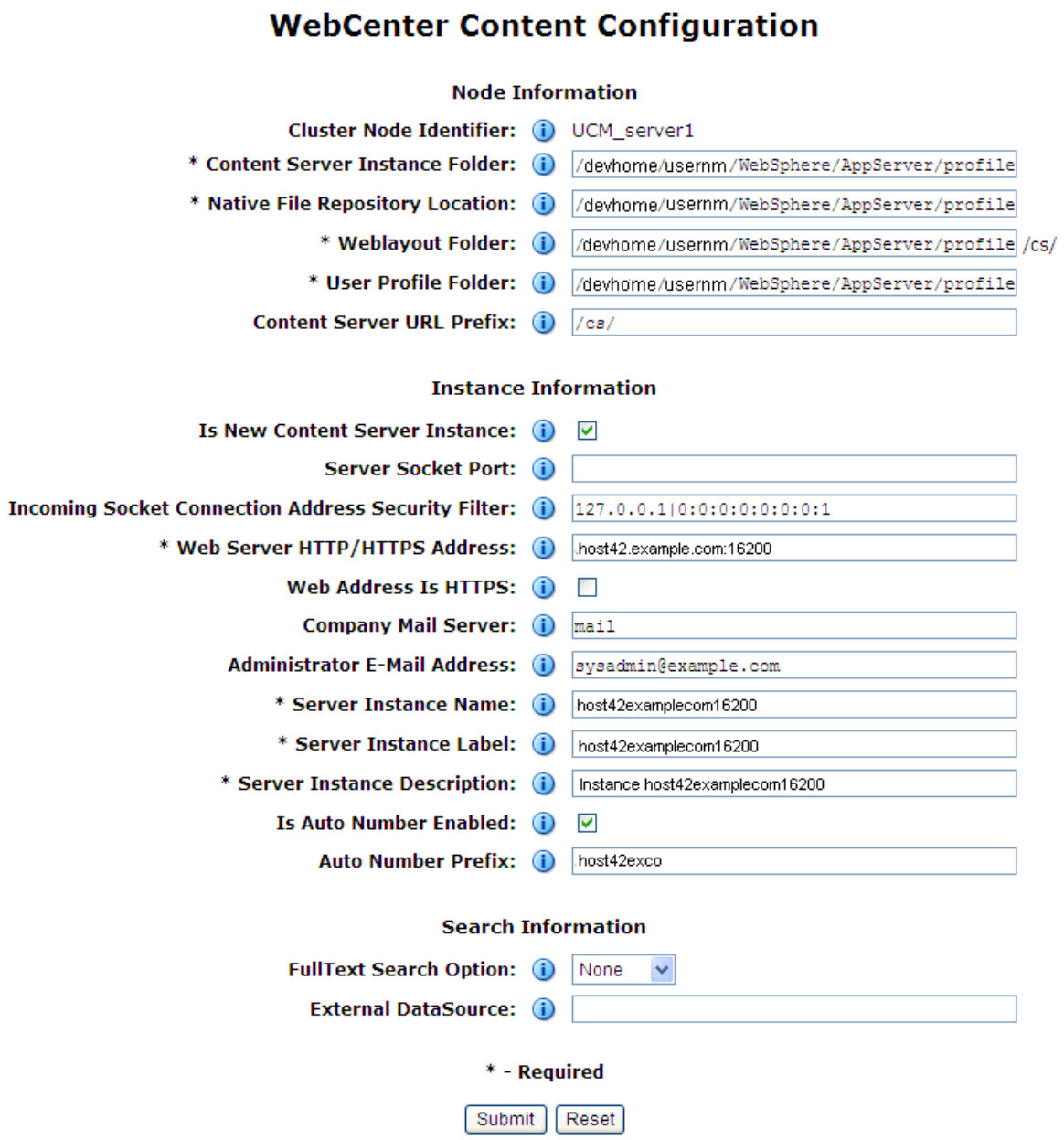
6.3.1 Mapping the weblayout Directory

The `weblayout` directory is the directory where WebCenter Content stores all the content checked into the server. This location is also where the content and other web assets are retrieved by the servlet or web server. WebCenter Content configures the mapping of the context root to a file system path at runtime in the servlet initializations. This configuration is done because the administrator can change the file system location of the `weblayout` directory during the initial configuration of Content Server, which happens the first time it is accessed.

For WebSphere Application Server, the `weblayout` directory is mapped to the `HttpRelativeWebRoot` value at runtime by appending the context root for WebCenter Content, `/cs/` by default, to the URL for the `weblayout` directory. You can locate this directory on a network file system to accommodate a large collection of content.

Figure 6–8 shows the postinstallation configuration page for Content Server. Before using Content Server, you need to make sure the context root for WebCenter Content is appended to the path for the `weblayout` directory, make any other configuration changes, and click **SUBMIT** to confirm the Content Server configuration. Then you need to and restart Content Server, as described in Section 6.4.1, "Starting or Restarting Content Server on IBM WebSphere."

Figure 6–8 Content Server Postinstallation Configuration



For information about the other values on the WebCenter Content Configuration page, see "Completing the Initial Configuration of Content Server" in the *Oracle WebCenter Content Installation Guide*.

6.3.2 Changing the Authentication Method for Oracle WebCenter Content Applications

If you want an Oracle WebCenter Content application to participate in single sign-on, you must specify `CLIENT-CERT` as the authentication method. By default, Oracle WebCenter Content applications specify `FORM` their authentication method. Unlike Oracle WebLogic Server, WebSphere Application Server *does not* support multiple, comma-separated authentication methods. You must change the authentication

method to CLIENT-CERT for any Oracle WebCenter Content application to participate in single sign-on.

To change the authentication method for an Oracle WebCenter Content application:

1. Locate the web.xml file for the application.

   For example, on a UNIX machine where WebSphere Application Server is installed, locate web.xml at

   ```
   WAS_HOME/profiles/profile_name/
      config/cells/cellName/applications/
      Oracle Universal Content Management - Content Server.ear/
      deployments/Oracle WebCenter Content - Content Server/cs.war/WEB-INF/web.xml
   ```

2. Copy web.xml to a temporary location.
3. Open web.xml in a text editor, and make these changes:
   a. Remove (or comment out) the following <login-config> element:

      ```
      <login-config>
       <auth-method>FORM</auth-method>
       <realm-name>idcauth</realm-name>
        <form-login-config>
          <form-login-page>/login/login.htm</form-login-page>
          <form-error-page>/login/error.htm</form-error-page>
        </form-login-config>
      </login-config>
      ```

   b. Replace it with the following <login-config> element:

      ```
      <login-config>
        <auth-method>CLIENT-CERT</auth-method>
      </login-config>
      ```

   c. Save the changes.
4. Redeploy the updated web.xml file:
   a. Open the IBM WebSphere Administrative Console, at https://*hostname*:*WC_Adminhost_port*/ibm/console; for example:

      ```
      https://host42.example.com:9002/ibm/console
      ```

   b. Log in as an administrator.
   c. Expand **Applications** and **Application Types** on the left of the console, and click **WebSphere enterprise applications**.
   d. Select **Oracle Universal Content Management - Content Server**, and click the **Update** button.
   e. Choose **Replace or add a single file**.
   f. Specify the path to the web.xml file you want to replace, starting with the name of the application's archive file (.war):

      ```
      cs.war/WEB-INF/web.xml
      ```

   g. Click **Next**.
   h. In the section **Specify the path to the file**, enter the full path to the web.xml file you updated in Step 3.
   i. Click **Next**, and follow through until the application is saved.

j. Click **OK**, and then click **Save Changes**.

Wait for a couple of minutes for the changes to be propagated.

k. To confirm the changes, navigate to the application's deployment descriptor, and view it.

Under **WebSphere Enterprise Applications**, choose **Oracle Universal Content Management - Content Server**, then **Manage Modules**, then **cs.war**, and then **View Deployment Descriptor**.

5. Restart the WebCenter Content application server, as described in Section 3.2.2, "Starting and Stopping Servers on IBM WebSphere."

6.4 Administering Oracle WebCenter Content Applications on IBM WebSphere

The following sections describe differences for performing some administration tasks for Oracle WebCenter Content Applications on IBM WebSphere instead of on Oracle WebLogic Server:

- Starting or Restarting Content Server on IBM WebSphere
- Logging In to WebCenter Content Server and Records
- Managing an Oracle WebCenter Content Cell and Servers from the IBM WebSphere Administrative Console
- Managing an Oracle WebCenter Content Cell, Servers, and Applications from Fusion Middleware Control

For information about Oracle Fusion Middleware administration on IBM WebSphere, see Chapter 3, "Managing Oracle Fusion Middleware on IBM WebSphere".

For information about administering Content Server in a WebCenter Content application server, see the *Oracle WebCenter Content System Administrator's Guide for Content Server*.

For information about administering an Inbound Refinery application server, see the *Oracle WebCenter Content Administrator's Guide for Conversion*.

For information about administering a Records application server, see the *Oracle WebCenter Content Administrator's Guide for Records*.

6.4.1 Starting or Restarting Content Server on IBM WebSphere

The Oracle WebCenter Content application (WebCenter Content), includes Oracle WebCenter Content Server (Content Server). You can start the WebCenter Content application server on IBM WebSphere with a profile or with Fusion Middleware Control. Then you can start a web browser and log in to Content Server. After you start the WebCenter Content application server, Content Server does not start until you access it for the first time.

To start Content Server on IBM WebSphere:

1. Start the WebCenter Content application server, as described in Section 3.2.2, "Starting and Stopping Servers on IBM WebSphere."
2. Go to the Content Server web interface, which is at this URL by default:

```
http://hostname:16200/cs
```

In the URL, *hostname* is the name of the machine (or host) on which Oracle WebCenter Content Server is running.

If a different port was configured for the WebCenter Content application server, specify that port instead of `16200`. The default SSL port is `16201`. For WebCenter Content, and Content Server, you can configure any port in the range `16200-16299`.

3. Log in to Content Server with the user name and password that you used to start the WebCenter Content application server.

To restart Content Server on IBM WebSphere:

1. Log out from the Content Server web interface.
2. Shut down the WebCenter Content application server, as described in Section 3.2.2, "Starting and Stopping Servers on IBM WebSphere."
3. Start the WebCenter Content application server, as described in Section 3.2.2, "Starting and Stopping Servers on IBM WebSphere."
4. Go to the Content Server web interface, which is at this URL by default:

   `http://`*`hostname`*`:16200/cs`

   In the URL, *hostname* is the name of the machine (or host) on which Oracle WebCenter Content Server is running.

   If a different port was configured for the WebCenter Content application server, specify that port instead of `16200`. The default SSL port is `16201`. For WebCenter Content, and Content Server, you can configure any port in the range `16200-16299`.

5. Log in to Content Server with the user name and password that you used to start the WebCenter Content application server.

6.4.2 Logging In to WebCenter Content Server and Records

Logging in to the Content Server web interface also logs you in to the Records web interface if WebCenter Content and Records are deployed to the same WebSphere cell. If you log in to Content Server from a browser and then go to the Records URL from another browser tab, you will not be prompted to log in to Records. The login credentials pass from Content Server to Records.

Logging out from Content Server also logs you out from Records.

This is different from logging in to Content Server and Records configured in the same Oracle WebLogic Server domain, where you need to log in to the web interface for each Managed Server separately.

6.4.3 Managing an Oracle WebCenter Content Cell and Servers from the IBM WebSphere Administrative Console

You can manage an Oracle WebCenter Content cell and its servers from the IBM WebSphere Administrative Console, which provides a web-based interface for managing the WebSphere Application Server environment. For more information, see Section 3.1.1, "Using the WebSphere Administrative Console."

6.4.4 Managing an Oracle WebCenter Content Cell, Servers, and Applications from Fusion Middleware Control

You can manage Oracle WebCenter Content applications (components) as well as their cell and servers from Oracle Enterprise Manager Fusion Middleware Control, which provides a web-based interface for monitoring and administering Oracle Fusion Middleware. For more information, see Section 3.1.2, "Using Oracle Enterprise Manager Fusion Middleware Control."

7

Managing Web Services on IBM WebSphere

Oracle Infrastructure Web Services and Oracle Web Services Manager are supported on IBM WebSphere, with some limitations. The tasks required to secure and administer Oracle Infrastructure Web services are described in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*. This chapter provides specific information for managing Oracle Fusion Middleware Web services on IBM WebSphere, and describes the limitations.

This chapter contains the following sections:

7.1 Configuring a Default Administrative User from the LDAP Directory

On WebSphere, Oracle Platform Security Services (OPSS) supports LDAP-based registries only; in particular, it does not support WebSphere's built-in file-based user registry. For information about configuring an LDAP registry and seeding the registry with users and groups required by Fusion Middleware components such as Oracle WSM, see Chapter 8, "Managing Oracle Fusion Middleware Security on IBM WebSphere.".

By default, the Oracle WSM Policy Manager uses the `wasadmin` administrative user to communicate with the server. If this user is not available in the LDAP, you must configure the policy manager to use a principle administrative user from the LDAP as described in the following procedure.

1. Configure the LDAP registry as described in "IBM WebSphere Identity Stores" on page 8-1 and restart the server.

> **Note:** The remaining steps in this procedure use the following sample primary user properties: `cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com` and `orcladmin-csf-key` for the `jndi.lookup.csf.key` that will be used for the administrator user access. The values for these properties will vary depending on your environment.

2. Update the credential store `cwallet.sso` file and the security role mappings using wsadmin commands as follows:

```
Opss.createCred (map='oracle.wsm.security', key='orcladmin-csf-key',
user='cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com', password='welcome1',
desc='wsm-pm admin user csf-key')

AdminApp.edit ('wsm-pm', '[-MapRolesToUsers [[policy.Updater
AppDeploymentOption.No AppDeploymentOption.No
cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com "" AppDeploymentOption.No
"user:cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com" "" ]]]']

AdminApp.edit('wsm-pm', '[ -MapRolesToUsers [[ policy.Accessor
AppDeploymentOption.No AppDeploymentOption.No
cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com "" AppDeploymentOption.No "
|user:cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com" "" ]]]' )

AdminApp.edit('wsm-pm', '[ -MapRolesToUsers [[ policy.User
AppDeploymentOption.No AppDeploymentOption.No
cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com "" AppDeploymentOption.No "
user:cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com" "" ]]]' )

AdminApp.edit('wsm-pm', '[ -MapRolesToUsers [[ policyViewer
AppDeploymentOption.No AppDeploymentOption.No
cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com "" AppDeploymentOption.No "
|user:cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com" "" ]]]' )

AdminConfig.save()

exit
```

> **Note:** The syntax for the `policyViewer` property differs from that of the other properties in that it does not include the separating period. Specifically, the syntax for these properties is `policy.Updater`,`policy.Accessor`,`policy.User`, `policyViewer`.

3. Restart the server.

7.2 Configuring Oracle WSM on IBM WebSphere

The following sections describe how to configure Oracle WSM and connect to the policy manager:

- Configuring Oracle WSM
- Connecting to the Oracle WSM Policy Manager

7.2.1 Configuring Oracle WSM

Oracle WSM is installed by default when you install Oracle Fusion Middleware SOA Suite or Oracle Application Development Runtime. For more information about installation, see Chapter 2, "Installing and Configuring Oracle Fusion Middleware on IBM WebSphere."

To configure Oracle Fusion Middleware in a new IBM WebSphere environment, you use a special version of the Oracle Fusion Middleware Configuration Wizard as

described in "Using the Configuration Wizard" in *Configuration Guide for IBM WebSphere Application Server*.

To configure Oracle WSM when you create or extend a cell using the Configuration Wizard, be sure to select the following options in the Add Products to Cell screen:

- **Oracle Enterprise Manager for WebSphere**
- **Oracle WSM Policy Manager**

If you plan to use asynchronous Web services, select **Oracle JRF WebServices Asynchronous services** also. For more information, see "Asynchronous Web Services" on page 7-6.

> **Note:** **Oracle JRF for WebSphere** is automatically selected as a dependency when you select any of the above products.

7.2.2 Connecting to the Oracle WSM Policy Manager

In a WebSphere environment, the Oracle WSM Policy Manager does not run on the same server as Oracle Enterprise Manager. Therefore, the Oracle WSM automatic discovery feature cannot locate and connect to an Oracle WSM Policy Manager. To connect to the policy manager, use the following procedure:

1. In the navigator pane of Enterprise Fusion Middleware Control, expand **WebSphere Cell** to view the cells.
2. Select the cell for which you want to configure the policy manager.
3. Right-click the name of the cell and from the menu select **Web Services** then **Platform Policy Configuration**.

   The Platform Policy Configuration page displays, as shown in Figure 7–1.

Figure 7–1 Platform Policy Configuration

4. Select the **Policy Accessor** tab.

   The Policy Accessor tab enables you to explicitly set a remote JNDI provider URL and corresponding csf-key credentials to access a Policy Manager on a remote server.

5. Click **Add** to define the remote JNDI provider.

   In the Add New Configure Property window, specify the following values:

a. In the Name field, enter the JNDI provider URL property as `java.naming.provider.url`.

b. In the Value field, enter the URL for the server on which the policy manager is running. For example:

```
corbaloc:iiop:hostname:rmiport
```

where *`hostname`* specifies the DNS name or IP address of the WebSphere server and *`rmiport`* specifies the port number on which the policy manager is running.

c. Click **OK**.

6. Click **Add** to define a corresponding csf-key credential property.

If the location of the Oracle WSM Policy Manager is provided in the `java.naming.provider.url` property, the `jndi.lookup.csf.key` provides the credential configuration.

Note: The csf-key that you specify in this step must match the csf-key specified for the Policy Manager administrative user in the credential store. For more information about adding an Oracle WSM Policy Manager administrative user to the credential store, see "Configuring a Default Administrative User from the LDAP Directory" on page 7-1.

In the Add New Configure Property window, specify the following values:

a. In the Name field, enter the name of the JNDI provider's csf-key credential property as `jndi.lookup.csf.key`.

b. In the Value field, enter the csf-key credentials.

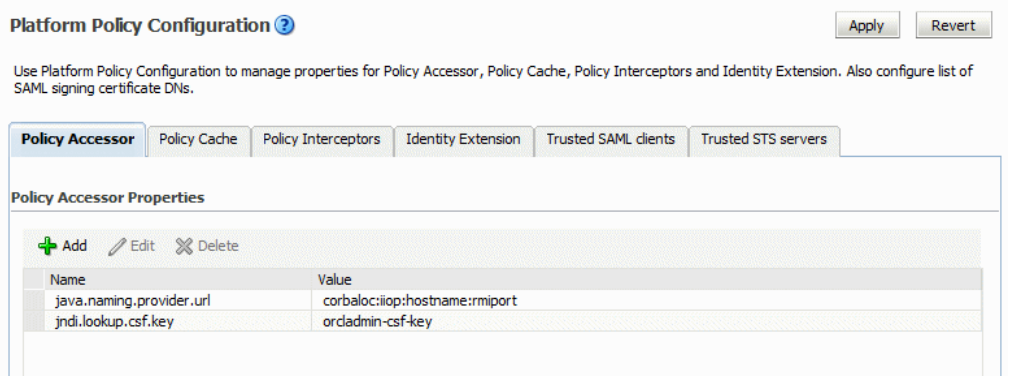
Because the Policy Manager is security enabled, the csf-key specifies the `java.naming.security.principal` and `java.naming.security.credentials` when using the JNDI URL to look up a Policy Manager.

For example, using the sample provided in "Configuring a Default Administrative User from the LDAP Directory" on page 7-1, the administrative user is `orcladmin` and the csf-key is `orcladmin-csf-key`.

c. Click **OK**.

Figure 7–2 shows the Policy Accessor tab with the `java.naming.provider.url` and `jndi.lookup.csf.key` property settings.

Figure 7–2 Policy Accessor Property Settings



For information about additional properties you can set on the Policy Accessor tab, see "Configuring Web Service Policy Retrieval" in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*.

7. Optionally, select the **Policy Cache** tab.

   The Policy Cache tab allows you to tune the behavior of the policy cache delay for Web service endpoints, which can help to avoid network calls and increase performance when fetching policies from a remote Oracle WSM Policy Manager.

8. To modify an existing policy cache property, select it and then click **Edit**. In the Edit Policy Cache Property window, you can edit the Value field to change the default amount for the property.

   You may want to edit the following property:

   - `cache.tolerance` – This ensures that the policy set retrieved from the Web service endpoint policy cache is the most current version (that is, it has not exceeded the `cache.tolerance` value). If it is determined that the policy set is stale, the updated policy set is retrieved from the Oracle WSM policy manager and refreshed in the Web service endpoint policy cache. The default is 60000 milliseconds (1 minute).

9. To add another property, click **Add**, and in the Add New Policy Cache Property window, specify the necessary values.

10. To delete an existing property, select it and then click **Delete**.

11. Click **Apply** to apply the property updates.

7.3 Differences and Restrictions When Developing Web Services Applications on IBM WebSphere

The following sections describe the differences when developing Web services applications on IBM WebSphere:

- High Availability
- Asynchronous Web Services
- JDeveloper

7.3.1 High Availability

Not all high availability (HA) features may be available at the same quality of service levels as WebLogic Server.

For example, Jython scripts are not available to configure the Java Object Cache in a clustered environment.

7.3.2 Asynchronous Web Services

Asynchronous Web services are supported on platforms other than WebLogic Server. For asynchronous Web services to function, the following JMS default queues must be present:

- oracle.j2ee.ws.server.async.DefaultRequestQueue
- oracle.j2ee.ws.server.async.DefaultResponseQueue
- oracle.j2ee.ws.server.async.DefaultRequestErrorQueue
- oracle.j2ee.ws.server.async.DefaultResponseErrorQueue
- weblogic.jms.XAConnectionFactory

To create these queues, you must configure Oracle JRF Asynchronous Web Services using the Oracle Fusion Middleware Configuration Wizard. You do so in the Add Products to Cell screen in the Configuration Wizard as described in "Configuring Oracle WSM" on page 7-2. Once you have created or extended a cell with this template, the JMS queues are available for use.

7.3.3 JDeveloper

When using JDeveloper, the remote Oracle WSM policy store on a WebSphere server is not available.

7.4 Differences and Restrictions When Managing Web Services Components on IBM WebSphere

The following sections describe the differences and restrictions for managing Web services components on IBM WebSphere:

7.4.1 Automatic Discovery of Oracle WSM Policy Manager

Automatic discovery of the Oracle WSM policy manager is not supported by third-party application servers, such as WebSphere. For details about connecting to the policy manager, see "Configuring Oracle WSM on IBM WebSphere" on page 7-2.

7.4.2 Web Services Atomic Transactions

Web Services Atomic Transactions (WSAT) are not supported and will result in runtime errors.

7.4.3 No Support for Native Web Services

Native Web services, such as those that are deployed to a stack other than the Oracle Infrastructure Web Services stack, are not exposed in the WSIL. Only the deployed Oracle Infrastructure Web Services are listed. The WSIL application is deployed on every server as part of the JRF template and the URI to access the application is /inspection.wsil. The wsil application uses basic HTTP authentication to ensure that only authorized users can access the list of Web services.

7.4.4 Reliable Messaging

WS-Reliable Messaging (WS-RM) is supported on IBM WebSphere with the following limitations:

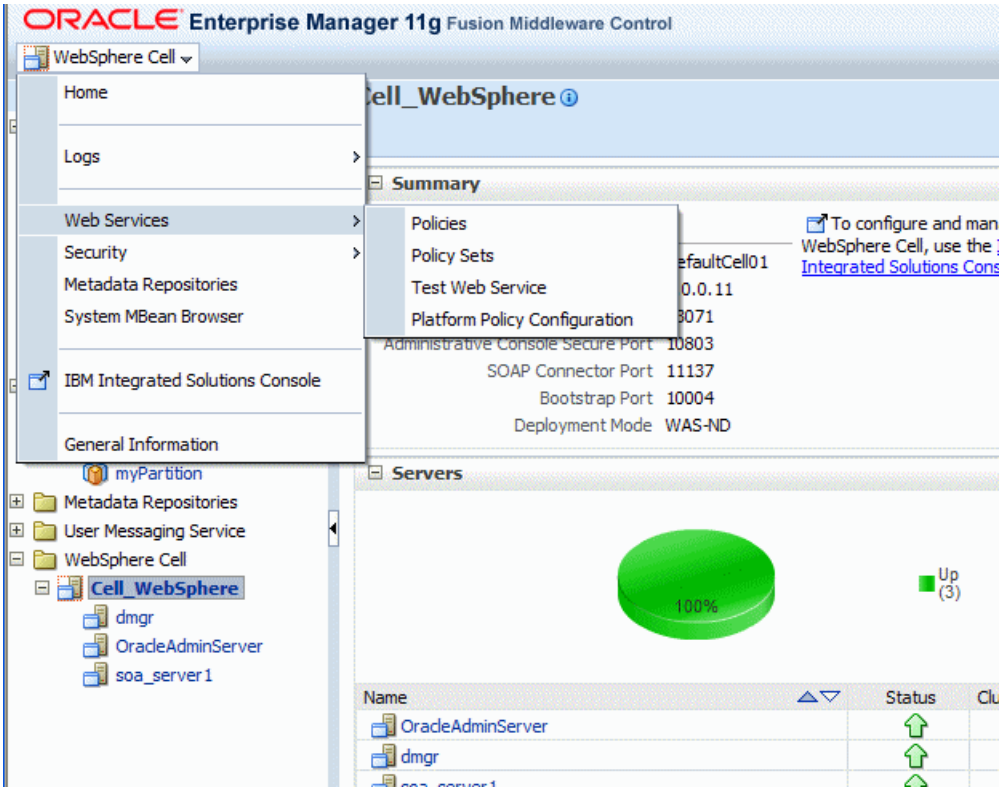
- WS-RM includes support for persistent database (DB) message store with Oracle databases only.
- WS-RM supports clustering only when Coherence is installed and available. This behavior is the same as WebLogic Server on all the platforms where Coherence is available.

7.4.5 Enterprise Manager Fusion Middleware Control

On IBM WebSphere, you access the Web services pages in Fusion Middleware Control using either of the following methods:

- From the main **WebSphere Cell** menu, select **Web Services**, then the desired Web services page, as shown in Figure 7–3.

Figure 7–3 Web Services Menu



- In the navigation pane, right-click on the target cell name, then select **Web Services**, then the desired Web services page.

The following limitations and differences apply when managing Web services using Fusion Middleware Control:

- You cannot view or manage Web services at the server level.
- The bulk policy attachment feature is not available.
- The registered sources and services, and publish to UDDI features are not available.
- The Application Deployment Summary page does not include the list of Web Services, or the Most Requested table.
- Native WebSphere Web services are not supported.
- The Usage Analysis page displays the WebSphere cell and server names.

7.5 Using the Web Services wsadmin Commands

The Web services wsadmin commands are identical to the custom Web services WebLogic Scripting Tool (WLST) commands provided for WebLogic Server. The Web services commands are grouped into two categories:

- WebServices—These commands consist of the Web service and client management commands, and the policy management commands. For a complete list of these commands, see "WebServices wsadmin Commands" on page 7-9.
- wsmManage—These commands consist of the policy set management commands, the import/export repository commands, and the Oracle WSM repository maintenance commands. For a complete list of these commands, see "wsmManage wsadmin Commands" on page 7-11.

> **Note:** Because the Oracle WSM Policy Manager is security enabled, you must pass Java system properties, such as username and password, when invoking wsadmin. For details about invoking wsadmin and using the wsadmin commands, see "Using the Oracle Fusion Middleware wsadmin Commands" on page 3-7

Refer to the following sections for more information:

- Executing the Web Services wsadmin Commands
- WebServices wsadmin Commands
- wsmManage wsadmin Commands

7.5.1 Executing the Web Services wsadmin Commands

To execute the wsadmin commands, you must prefix each command with the category name. That is, each command in the WebServices category must be preceded by `WebServices`, and each command in the wsmManage category must be preceded with `wsmManage`. For example:

- To execute a command in the WebServices category, such as the `listWebServices()` command, enter the following:

```
wsadmin>WebServices.listWebServices(None, None, 'true')
```

```
/NonTLRCell/OracleAdminServer/j2wbasicPolicy :
        moduleName=j2wbasicPolicy, moduleType=web,
serviceName=WssUsernameService
        enableTestPage: true
        enableWSDL: true

                JRFWssUsernamePort
http://host.us.oracle.com:9002/j2wbasicPolicy/WssUsername
                enable: true
                enableREST: false
                enableSOAP: true
                maxRequestSize: -1
                loggingLevel: NULL
                wsat.flowOption: NEVER
                wsat.version: DEFAULT
                security : oracle/wss_username_token_service_policy,
enabled=true, effective=true
                addressing : oracle/wsaddr_policy, enabled=true
                (global) security : oracle/binding_authorization_permitall_
policy, enabled=true
                        /policysets/global/app-only-web-service-policies :
Application("j2wbasicPolicy")
                Attached policy or policies are valid; endpoint is secure.
```

- To execute a command in the wsmManage category, such as the `listPolicySets()` command, enter the following:

```
wsadmin>wsmManage.listPolicySets()

Global Policy Sets in Repository:
  all-cells-default-web-service-policies
  app-only-web-service-policies
```

7.5.2 WebServices wsadmin Commands

The following table identifies the WebServices management wsadmin commands that are supported on WebSphere, and provides links to the reference documentation in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*. Sample procedures for using the commands are described in the following chapters in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*:

- Administering Web Services
- Managing Web Service Policies
- Attaching Policies to Web Services

> **Note:** You can use these commands as described in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* and *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*. However, in a WebSphere environment, you must execute the commands as described in "Executing the Web Services wsadmin Commands" on page 7-8.

Table 7–1 WebServices wsadmin Commands Supported on IBM WebSphere

Command	Description
listWebServices	List the Web service information for an application, composite, or cell.
listWebServicePorts	List the Web service ports for a Web service application or SOA composite.
listWebServiceConfiguration	List Web services and port configuration for an application or SOA composite.
listWebServiceClients	List Web service client information for an application, SOA composite, or cell.
listWebServiceClientPorts	List Web service client ports information for an application or SOA composite.
listWebServiceClientStubProperties	List Web service client port stub properties for an application or SOA composite.
setWebServiceConfiguration	Set or change the Web service port configuration for a Web service application or SOA composite.
setWebServiceClientStubProperty	Set, change, or delete a single stub property of a Web service client port for an application or SOA composite.
setWebServiceClientStubProperties	Configure the set of stub properties of a Web service client port for an application or SOA composite.
listAvailableWebServicePolicies	Display a list of all the available Oracle Web Services Manager (WSM) policies by category or subject type.
listWebServicePolicies	List Web service port policy information for a Web service in an application or SOA composite.
listWebServiceClientPolicies	List Web service client port policies information for an application or SOA composite.
attachWebServicePolicy	Attach a policy to a Web service port of an application or SOA composite.
attachWebServicePolicies	Attach multiple policies to a Web service port of an application or SOA composite.
attachWebServiceClientPolicy	Attach an Oracle WSM policy to a Web service client port of an application or SOA composite.
attachWebServiceClientPolicies	Attach multiple policies to a Web service client port of an application or SOA composite.
enableWebServicePolicy	Enable or disable a policy attached to a port of a Web service application or SOA composite.
enableWebServicePolicies	Enable or disable multiple policies attached to a port of a Web service application or SOA composite.
enableWebServiceClientPolicy	Enable or disable a policy of a Web service client port of an application or SOA composite.
enableWebServiceClientPolicies	Enable or disable multiple policies of a Web service client port of an application or SOA composite.
detachWebServicePolicy	Detach an Oracle WSM policy from a Web service port of an application or SOA composite.
detachWebServicePolicies	Detach multiple Oracle WSM policies from a Web service port of an application or SOA composite.

Table 7–1 (Cont.) WebServices wsadmin Commands Supported on IBM WebSphere

Command	Description
detachWebServiceClientPolicy	Detach a policy from a Web service client port of an application or SOA composite.
detachWebServiceClientPolicies	Detach multiple policies from a Web service client port of an application or SOA composite.
setWebServicePolicyOverride	Configure the Web service port policy override properties of an application or SOA composite.

7.5.3 wsmManage wsadmin Commands

The following table identifies the wsmManage commands that are supported on WebSphere, and provides links to the reference documentation in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*. Sample procedures for using these commands are described in the following chapters in *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*:

- Creating and Managing Policy Sets
- Managing Application Migration Between Environments
- Maintaining the Oracle WSM MDS Repository

Note: You can use these commands as described in *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* and *Oracle Fusion Middleware Security and Administrator's Guide for Web Services*. However, in a WebSphere environment, you must execute the commands as described in "Executing the Web Services wsadmin Commands" on page 7-8.

Table 7–2 wsmManage Commands Supported on IBM WebSphere

Command	Description
beginRepositorySession	Begin a session to modify the Oracle MDS repository.
commitRepositorySession	Write the contents of the current session to the Oracle MDS repository.
abortRepositorySession	Abort the current Oracle MDS repository modification session, discarding any changes that were made to the repository during the session.
describeRepositorySession	Describe the contents of the current repository session.
attachPolicySet	Attach a policy set to the specified resource scope.
attachPolicySetPolicy	Attach a policy to a policy set using the policy's URI.
detachPolicySetPolicy	Detach a policy from a policy set using the policy's URI.
clonePolicySet	Clone a new policy set from an existing policy set.
createPolicySet	Create a new, empty policy set.
deletePolicySet	Delete a specified policy set.
deleteAllPolicySets	Delete all or selected policy sets from within the Oracle WSM repository.
displayPolicySet	Display the configuration of a specified policy set.
enablePolicySet	Enable or disable a policy set.
enablePolicySetPolicy	Enable or disable a policy attachment for a policy set using the policy's URI.

Table 7–2 (Cont.) wsmManage Commands Supported on IBM WebSphere

Command	Description
listPolicySets	Lists the policy sets in the repository.
modifyPolicySet	Specify an existing policy set to be modified in the current session.
setPolicySetPolicyOverride	Add a configuration override to a policy reference in the current policy set.
setPolicySetConstraint	Specify a run-time constraint value for a policy set selected within a session.
setPolicySetDescription	Specify a description for the policy set selected within session.
validatePolicySet	Validate existing policy set in the repository or in a session.
migrateAttachments	Migrates direct policy attachments to global policy attachments if they are identical.
importRepository	Import a set of documents from a supported ZIP archive file into the repository. You can provide the location of a file that describes how to map physical information from the source environment to the target environment.
exportRepository	Export a set of documents from the repository into a supported ZIP archive. If the specified archive already exists, you can choose whether to overwrite the archive or merge the documents into the existing archive.
upgradeWSMPolicyRepository	Upgrade the Oracle WSM predefined policies stored in the Oracle MDS repository with any new predefined policies that are provided in the latest installation of the Oracle Fusion Middleware software.
resetWSMPolicyRepository	Delete the existing policies stored in the Oracle MDS repository and refresh it with the latest set of predefined policies that are provided in the new installation of the Oracle Fusion Middleware software.

8

Managing Oracle Fusion Middleware Security on IBM WebSphere

This chapter contains information about managing Oracle Fusion Middleware security on IBM WebSphere, and it explains the particularities of some Oracle Platform Security Services (OPSS) features on that platform.

OPSS is a security platform that can be used to secure applications deployed in any of the supported platforms or in standalone applications.

Only topics that apply specifically to IBM WebSphere are included in this chapter; those that apply uniformly to all platforms are not described here, but can be found in *Oracle Fusion Middleware Application Security Guide*.

This chapter contains the following sections:

8.1 IBM WebSphere Identity Stores

On IBM WebSphere, OPSS supports LDAP-based registries only; in particular, it does not support WebSphere's built-in file-based user registry.

For information about the list of LDAP authenticators supported for Oracle Fusion Middleware, visit
`http://www.oracle.com/technology/software/products/ias/files/fusion_certification.html`

For the special configuration required for the Open LDAP 2.2, see *Oracle Fusion Middleware Application Security Guide*.

The configuration and seeding of a repository is explained in the following sections:

- Configuring a Registry
- Seeding a Registry

8.1.1 Configuring a Registry

The configuration of an LDAP registry on IBM WebSphere is accomplished with the command `configureIdentityStore`, an online administration command with the following syntax:

```
wsadmin> Opss.configureIdentityStore(propsFileLoc="fileLocation")
```

`propsFileLoc` specifies the location of the file that contains the property settings for the identity LDAP identity store. This command modifies the configuration file `jps-config.xml` to include the specifications in the property file.

After running Opss.configurIdentityStore, the server must be restarted.

The following properties are required and must be specified in property settings file:

- ldap.host
- ldap.port
- admin.id
- admin.pass
- idstore.type
- user.search.bases
- user.id.map
- group.id.map
- group.member.id.map
- group.search.bases
- primary.admin.id

The following list includes optional properties specific to a IBM WebSphere registry:

- group.filter
- user.filter

The following sample illustrates the property settings for an Oracle Directory Server Enterprise Edition identity store:

```
user.search.bases=cn=Users,dc=us,dc=oracle,dc=com
group.search.bases=cn=Groups,dc=us,dc=oracle,dc=com
subscriber.name=dc=us,dc=oracle,dc=com
ldap.host=stamw10.us.oracle.com
ldap.port=3060
# admin.id must be the full DN of the user in the LDAP
admin.id=cn=orcladmin
admin.pass=welcome1
user.filter=(&(uid=%v)(objectclass=person))
group.filter=(&(cn=%v)(objectclass=groupofuniquenames))
user.id.map=:uid
group.id.map=:cn
group.member.id.map=groupofuniquenames:uniquemember
ssl=false
```

```
# primary.admin.id indicates the user you want to be the primary
# administrative user on WebSphere. It should be a user under user.search.bases.
# later you need to use this user's user name and password to manage or
# start/stop the server.
primary.admin.id=orcladmin
# optional, default to "OID"
idstore.type=IPLANET
# other, optional identity store properties can be configured in this file.
username.attr=cn
```

The list of valid identity store types is the following:

- OID
- IPLANET
- OVD
- ACTIVE_DIRECTORY
- OPEN_LDAP

8.1.2 Seeding a Registry

Some Oracle Fusion Middleware components require that certain users and groups be present in the IBM WebSphere identity store. To ensure that this requirement is met, use any tools to seed the required data; in particular, you can use an LDIF file and the LDAP utility `bulkload` to load users and groups into the identity store. Here is a sample LDIF file:

```
dn: cn=OracleSystemUser,dc=com
userPassword: welcome1
sn: OracleSystemUser
cn: OracleSystemUser
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top

dn: cn=OracleSystemGroup,dc=com
cn: OracleSystemGroup
objectclass: groupOfUniqueNames

dn: cn=Administrators,dc=com
cn: Administrators
objectclass: groupOfUniqueNames

dn: cn=SystemMDBRole,dc=com
cn: SystemMDBRole
objectclass: groupOfUniqueNames
uniquemember: cn=OracleSystemUser,dc=com
```

8.2 Configuring the Trust Association Interceptor

HTTP clients can pass identity information to WebSphere Application Server using the Trust Association Interceptor (TAI). OPSS uses TAI as the asserter that intercepts calls coming into WebSphere cells to support identity propagation across containers and cells.

To configure TAI, proceed as follows:

1. Login to the IBM WebSphere Administrative Console.
2. Select **Security > Click Global Security**.
3. In the opened page, navigate to **Authentication**.
4. Expand **Web** and **SIP** security, and click **Trust Association**.
5. Check the box **Enable Trust Association** and save your changes.
6. Return to the Trust Association page and click **Additional Properties > Interceptors**.
7. Click **New**.
8. In the **Interceptor Class Name** box, enter the following string:

   ```
   oracle.security.jps.was.providers.trust.TrustServiceAsserterTAI
   ```

   This class is packaged in the JAR file `jps-was.jar`.
9. Save your changes.

8.3 Migrating Policies at Deployment

The migration of application policies at deployment is controlled by several parameters configured in the file `META-INF/opss-application.xml`. For an example of this file, see Sample opss-application File. To reassociate the policy store after deployment, see Reassociating Policies with reassociateSecurityStore.

The supported parameters, including configuration examples, are explained in the following sections:

- jps.policystore.migration
- jps.policystore.applicationid
- jps.policystore.removal

Note that the following parameters are not supported on IBM WebSphere:

```
JpsApplicationLifecycleListener
Jps.apppolicy.idstoreartifact.migration
Jps.policystore.migration.validate.principal
```

8.3.1 jps.policystore.migration

This parameter specifies whether the migration should take place, and, when it does, whether it should merge with or overwrite matching policies present in the target store.

On IBM WebSphere, it is configured as illustrated in the following fragment:

```
<service type="POLICY_STORE">
 <property name="jps.policystore.applicationid" value="stripeid" />
 <property name="jps.policystore.migration" value="overwrite" />
 <property name="jps.policystore.removal" value="off" />
</service>
```

For more details about this parameter, see *Oracle Fusion Middleware Application Security Guide*.

8.3.2 jps.policystore.applicationid

This parameter specifies the target stripe into which policies are migrated.

On IBM WebSphere, it is configured as illustrated in the following fragment:

```
<service type="POLICY_STORE">
 <property name="jps.policystore.applicationid" value="stripeid" />
 <property name="jps.policystore.migration" value="overwrite" />
 <property name="jps.policystore.removal" value="off" />
</service>
```

For more details about this parameter, see *Oracle Fusion Middleware Application Security Guide*.

8.3.3 jps.policystore.removal

This parameter specifies whether the removal of policies at undeployment should *not* take place.

On IBM WebSphere, it is configured as illustrated in the following fragment:

```
<service type="POLICY_STORE">
 <property name="jps.policystore.applicationid" value="stripeid" />
 <property name="jps.policystore.migration" value="overwrite" />
 <property name="jps.policystore.removal" value="off" />
</service>
```

For more details about this parameter, see *Oracle Fusion Middleware Application Security Guide*.

8.4 Migrating Credentials at Deployment

The migration of application credentials at deployment is controlled by a parameter configured in the file `META-INF/opss-application.xml`. For an example of this file, see Sample opss-application File.

The supported parameter, including a configuration example, are explained in the following section:

- jps.credstore.migration

Note that the following parameter is not supported on IBM WebSphere:

`jps.ApplicationLifecycleListener`

8.4.1 jps.credstore.migration

This parameter specifies whether the migration should take place, and, when it does, whether it should merge with or overwrite matching credentials present in the target store.

On IBM WebSphere, it is configured as illustrated in the following fragment:

```
<service type="CREDENTIAL_STORE">
 <property name="jps.credstore.migration" value="overwrite" />
</service>
```

Setting `jps.credstore.migration` to `overwrite` requires that the system property `jps.app.credential.overwrite.allowed` be set to `true`.

For more details about this parameter, see *Oracle Fusion Middleware Application Security Guide*.

8.5 Reassociating Policies with reassociateSecurityStore

For complete details about the scrip `reassociateSecurityStore` to reassociate the policy store, see *Oracle Fusion Middleware Application Security Guide*. Since this script is likely to run for some time, to avoid exceptions, one may need to reset the default connection to the server timeout to an appropriate larger value.

8.6 Deployment Mode

On IBM WebSphere, deployment is supported *only* in online mode; no offline deployment is supported.

8.7 Configuring the JpsFilter and the JpsInterceptor

On IBM WebSphere, both the JpsFilter and the JpsInterceptor must be manually configured.

For the properties supported and configuration examples, see *Oracle Fusion Middleware Application Security Guide*.

8.8 Using System Variables in Code Source URLs

The system variables `oracle.deployed.app.dir` and `oracle.deployed.app.ext` can be used to specify a URL independent of the platform. For a configuration example using these variables, see *Oracle Fusion Middleware Application Security Guide*.

8.9 Sample opss-application File

The following sample illustrates the contents of the `opss-application.xml` file.

```
<?xml version="1.0" encoding="UTF-8" standalone='yes'?>
<opss-application
xmlns="http://xmlns.oracle.com/oracleas/schema/11/opss-application-11_1.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/oracleas/schema/11/opss-application-11
_1.xsd" schema-major-version="11" schema-minor-version="1">
  <services>
    <service type="POLICY_STORE">
      <property name="jps.policystore.applicationid" value="stripeid" />
      <property name="jps.policystore.migration" value="MERGE" />
    </service>
    <service type="CREDENTIAL_STORE">
      <property name="jps.credstore.migration" value="MERGE" />
    </service>
  </services>
</opss-application>
```

8.10 About the File web.xml

The element `<auth-method>` in a `web.xml` file is WebLogic-specific and not supported on IBM WebSphere; if found, it must be replaced with the equivalent functionality supported for IBM WebSphere's `web.xml` files.

8.11 Executing Common Audit Framework wsadmin Commands

To run audit commands, provided by Oracle Fusion Middleware's Common Audit Framework, you need to do the following:

1. Start the Oracle Fusion Middleware wsadmin command-line shell.
2. Prefix the audit commands with the keyword `Audit`. For example:

```
wsadmin> Audit.getAuditPolicy()
wsadmin> Audit.setAuditPolicy()
```

For details about the audit commands, see the *Oracle Fusion Middleware Application Security Guide*.

9

Managing OAM Identity Assertion on IBM WebSphere

Oracle Access Manager Identity Assertion Provider for IBM WebSphere can be used to provide authentication and single sign-on with Oracle Access Manager 10*g* (10.1.4.3) or 11g.

> **Note:** IBM WebSphere is shorthand for IBM WebSphere Application Server. For more information, see "Supported IBM WebSphere Application Servers" on page 1-2.

This chapter includes the following topics:

9.1 Introduction to OAM Identity Assertion on IBM WebSphere

Oracle Access Manager Identity Assertion Provider is part of Oracle Fusion Middleware. Oracle provides an Identity Assertion Provider for IBM WebSphere that can be used to intercept and validate OAM sessions and generate IBM WebSphere-specific sessions.

IBM WebSphere allows Single Sign On (SSO) with external authenticators by using the Trust Association Interceptor (TAI). TAI interfaces provide mechanisms for external authenticators to perform user authentication and then assert the identity to IBM WebSphere. Oracle Access Manager Identity Assertion Provider for IBM WebSphere uses the TAI interface to assert the user identity from the OAM session to IBM WebSphere. Upon receiving user identity information from the Identity Assertion Provider, IBM WebSphere queries the existence of the user in the user registry.

Oracle Access Manager Identity Assertion Provider for IBM WebSphere needs a valid OAM session for asserting the user identity to IBM WebSphere. Typically this is achieved by using an IBM HTTP Server (IHS) reverse proxy to front-end IBM WebSphere. OAM WebGate is installed on the IHS proxy and used to authenticate users against Oracle Access Manager. WebGate generates an OAM session token upon successfully authenticating a user. The IHS proxy then forwards this session token to IBM WebSphere. The Identity Assertion Provider intercepts the request and asserts the user identity from the session token for IBM WebSphere.

The Identity Assertion Provider provides identity assertion using either the HTTP Cookie or HTTP Request Headers. Accordingly, the IAP can be configured for Cookie based assertion or header based assertion.

- Cookie-based Assertion: Is based on OAM Session Token (ObSSOCookie). In this configuration, the Identity Assertion Provider checks availability of ObSSOCookie and validates it. On successful validation, user identity in the session cookie is asserted to IBM WebSphere.
- Header-based Assertion: Is based on HTTP Request Header. In this configuration, the Identity Assertion Provider checks availability of a particular (configurable) request header in the request. If available, the user identity within the header is asserted to IBM WebSphere.

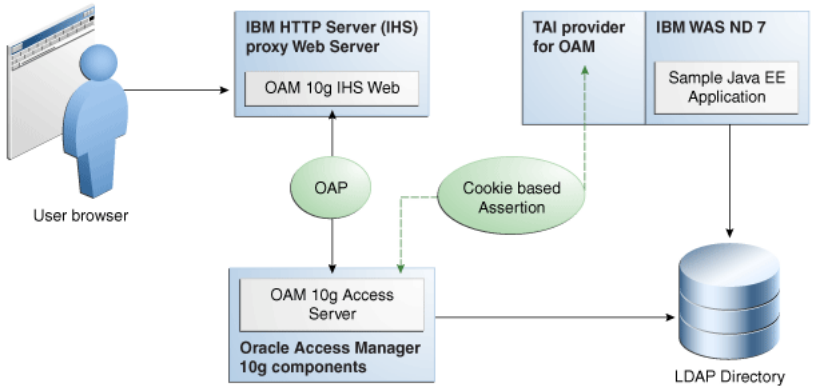
For more information, see the following topics:

- Scenario 1: Oracle Access Manager 10g (10.1.4.3) with the IAP on IBM WebSphere
- Scenario 2: OAM 11g with the IAP and IBM WebSphere

9.1.1 Scenario 1: Oracle Access Manager 10*g* (10.1.4.3) with the IAP on IBM WebSphere

This scenario describes a Java EE application that relies on Oracle Access Manager 10*g* (10.1.4.3) for authentication and authorization of its users. This application has been deployed on IBM WebSphere and can use the Identity Assertion Provider to provide SSO with Oracle Access Manager 10*g* (10.1.4.3).

Figure 9–1 Components and Process Flow with OAM 10g (10.1.4.3) and the IAP



Process overview: Identity Assertion on IBM WebSphere

1. Browser to IHS Proxy Web Server: User accesses the IBM WebSphere resource using the proxy IHS host and port, which triggers the 10*g* (10.1.4.3) WebGate installed on IHS Web server to authenticate and authorize the user.
2. WebGate to Access Server: WebGate communicates with OAM 10*g* (10.1.4.3) Access Server using Oracle Access Protocol (OAP). Access Server checks the Policy Store to locate any policies protecting the requested resource. WebGate through Access Server collects credential information from the user based on the Authentication Scheme specified and then validates whether the user can be authenticated. On successful authentication, WebGate through Access Server authorizes the user to access the requested resource on the IHS Web server. Additionally, WebGate sets authorization headers in the request as specified in the OAM Policy.
3. Web Server to IBM WebSphere: IHS Web Server acts as a proxy for IBM WebSphere and forwards the request to IBM WebSphere after successful authorization by OAM 10*g* (10.1.4.3) WebGate. IHS Web Server will also forward the HTTP Cookies and Request Headers set in the request to the IBM WebSphere.

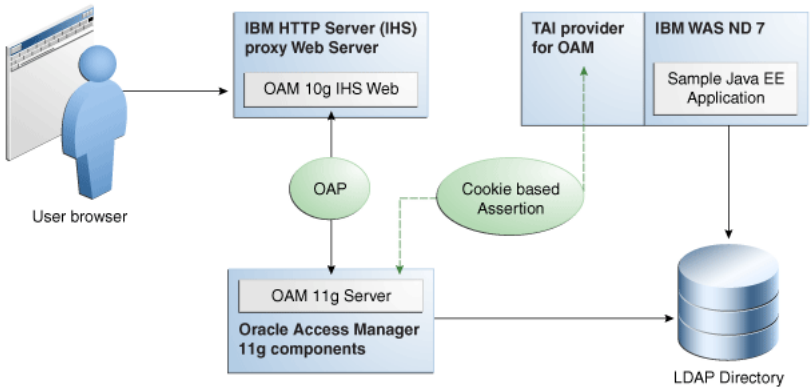
   Requests are intercepted at IBM WebSphere by OAM IAP. The TAI of OAM then validates the Cookie and HTTP Header. OAM IAP communicates with 10*g* (10.1.4.3) Access Server for Cookie-based assertions, to validate the session token and retrieve user information for the session. The TAI asserts this user identity to IBM WebSphere.

   IBM WebSphere checks for the existence of user in the user registry (configured LDAP instance) supplied by the OAM IAP. If the user is found, the assertion is successful. IBM WebSphere does not check for or request user's password in this scenario.
4. SSO Logout: See "Configuring SSO Logout for OAM IAP for IBM WebSphere" on page 9-20.

9.1.2 Scenario 2: OAM 11g with the IAP and IBM WebSphere

This scenario describes a Java EE application that relies on Oracle Access Manager 11g for authentication and authorization of its users. The Java EE application is deployed on IBM WebSphere to use the OAM IAP for IBM WebSphere for integrating the SSO with Oracle Access Manager 11g.

Figure 9–2 Components and Process Flow with OAM 11g and the IAP



Process overview: Identity Assertion with Oracle Access Manager 11g

1. Browser to IHS Proxy Web Server: The user accesses the resource (Sample Application on IBM WebSphere) using the proxy IHS host and port, which triggers the OAM 10*g* (10.1.4.3) WebGate installed to authenticate and authorize the user.
2. OAM 10*g* (10.1.4.3) IHS WebGate communicates with OAM 11g Server across the Oracle Access Protocol (OAP).

   OAM 11g Server checks its policy store to locate policies protecting the resource.

   WebGate and OAM 11g Server collect credentials from the user based on the authentication scheme specified in the policy, and the OAM 11g Server validates if the user can be authenticated.

   On successful authentication, WebGate and OAM Server authorize the user before access to the requested resource on the IHS Web server is granted. WebGate sets authorization headers in the request as specified in the OAM policy.
3. Web Server to IBM WebSphere: IHS Web Server acts as a proxy for IBM WebSphere and forwards the request to IBM WebSphere after successful authorization by OAM 10*g* (10.1.4.3) WebGate. IHS Web Server also forwards to IBM WebSphere the HTTP Cookies and Request Headers set in the request.

   Requests are intercepted at IBM WebSphere by OAM IAP. The TAI for OAM then validates the Cookie or HTTP Header. OAM IAP communicates with OAM 11g Server for Cookie-based assertions, to validate the session token, and retrieve user information for the session. TAI is responsible for asserting this user identity to IBM WebSphere.

   IBM WebSphere checks the existence of the user (supplied by the OAM IAP) in its user registry (configured LDAP instance). If user is found in the user registry, the assertion is successful. IBM WebSphere does not request nor check the user's password in this scenario.
4. SSO Logout: See "Configuring SSO Logout for OAM IAP for IBM WebSphere" on page 9-20.

9.2 Installing Components for the Oracle Access Manager IAP for IBM WebSphere

This section outlines the tasks you must perform to enable OAM Identity Assertion with IBM WebSphere.

The Oracle Access Manager IAP for IBM WebSphere is available as part of Oracle Fusion Middleware suite for IBM WebSphere. The IAP for IBM WebSphere jar is located at:

```
MW_HOME/oracle_common/modules/oracle.oamprovider_11.1.1/
    OAMTrustAssociationInterceptor.jar
```

Oracle Access Manager IAP for IBM WebSphere configuration file is located at:

```
MW_HOME/oracle_common/modules/oracle.oamprovider_11.1.1/
    domain_config_was/oamtai.xml
```

> **Note:** Oracle Access Manager 10*g* (10.1.4.3) components and installation differs from Oracle Access Manager 11g components and installation. However, all other component installation tasks are the same.

Task overview: Installing components for IBM WebSphere, OAM, and the IAP

1. Install and set up IBM WebSphere as described in Chapter 2, "Installing and Configuring Oracle Fusion Middleware on IBM WebSphere."
2. IBM HTTP Server 7.x can be used as a reverse proxy in front of IBM WebSphere.

   > **Note:** For IBM HTTP Server 7.x, use IHS22 WebGate package.

3. Oracle Access Manager: Install either:
   - OAM 10*g* (10.1.4.3): As described in the *Oracle COREid Access and Identity Installation Guide* and includes:

     10*g* (10.1.4.3) Identity Server
     10*g* (10.1.4.3) Access Server
     10*g* (10.1.4.3) Policy Manager
     10*g* (10.1.4.3) Web Components for OHS 11g Web Server: Web Pass, Policy Manager and Web Gate)

   - OAM 11g: As described in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*, which includes:

     Oracle Access Manager 11g (11.1.1.3.0)
     Oracle Identity Manager 11g (11.1.1.3.0
     Oracle WebLogic Server

4. WebGate: Required whether you use OAM 10*g* (10.1.4.3) or OAM 11g, and can be installed after provisioning as described later in this chapter.

9.3 Introduction to the Oracle Access Manager 10*g* (10.1.4.3) Configuration Tool

This section introduces OAMCfgTool (oamcfgtool.jar) is a platform-agnostic configuration tool for use with Oracle Access Manager 10*g* (10.1.4.3). Skip this topic if you have OAM 11g deployed.

> **See Also:** *Oracle Fusion Middleware Application Security Guide* for more information on OAMCfgTool

OAMCfgTool is a command-line utility provided to automatically run a series of scripts and set up policies. OAMCfgTool requires a set of parameters as inputs to create the required form-based authentication scheme, policy domain, access policies, and a WebGate profile for the Identity Asserter for single sign-on for IBM WebSphere.

Note: OAMCfgTool requires JRE 1.5 or 1.6. Internationalized login forms for Fusion Middleware applications are supported with the policies protecting those applications.

With OAM 10*g* (10.1.4.3) deployed, if you do not use the OAM Config Tool you must manually create the host-identifier, authentication schemes, and OAM policy manually using the Access System Console, as described in the *Oracle Access Manager Access Administration Guide.*

Example 9–1 a sample template for the configuration file for creating the required artifacts for the OAM IAP for IBM WebSphere. Additional information follows the example.

Example 9–1 Sample URIs_config File for OAMCfgTool and the IAP for IBM WebSphere

```
-- Template-starts --
##################################
#
# OAM-WAS Integration using OAM IAP
#
##################################
protected_uris

##################################
#Resources protected with default authentication scheme
/webcenter/adfAuthentication

##################################
public_uris
##################################
#Public Policy required for Cookie Based Assertion
Cookie Based Assertion
/Authen/SSOToken
-- Template-ends --
```

Example 9–2 illustrates a sample of the command-line syntax for OAMCfgTool when configuring artifacts for OAM 10*g* (10.1.4.3) and the IAP for IBM WebSphere.

Example 9–2 OAMCfgTool Syntax Configures Artifacts for OAM 10g (10.1.4.3) IAP

```
(echo ldappwdjava -jar oamcfgtool.jar
mode=CREATE app_domain=OAMPolicy_for_WAS-IAP
```

```
uris_file=/path-to-template-config-file
web_domain=host-id-name
ldap_host=wxyz
ldap_port=6633
ldap_userdn=orcladmin
ldap_base=ldap-base-dn
oam_aaa_host=abcd
oam_aaa_port=7789
oam_aaa_mode=open
log_file=OAMCfg_date.log
log_level=INFO
output_ldif_file=<LDIF_filename>
-noprompt
```

The above sample command produces the following artifacts:

- OAMPolicy_for_WAS-IAP, OAM Policy for protecting IBM WebSphere resources specified under protected_uris and public_uris
- OraDefaultAnonAuthNScheme, Anonymous Authentication Scheme used by OAMPolicy_for_WAS-IAP
- OraDefaultFormAuthNScheme, Form Authentication Scheme used by OAMPolicy_for_WAS-IAP
- Other OAM authentication scheme configuration

For a known resource, the public URI policy needs a Return Attribute in the Authorization Actions for Cookie-based assertion, as shown in Table 9–1. In this case, the return name OAM_REMOTE_USER is not configurable in oamtai.xml.

Table 9–1 Authorization Actions for "Cookie-based Assertion" in Public URI Policy

Type	Name	Return Attribute
HeaderVar	OAM_REMOTE_USER	uid

To enable Header-based assertion, you must set the Return Attribute in Authorization Actions of the Resource (protected_uris) protection policy. With Header-based Assertion, the return name OAM_REMOTE_USER is configurable in the oamtai.xml file and you must ensure that the Header-based Assertion section is uncommented.

Table 9–2 Authorization Actions for "Header Based Assertion" in Protected URI Policy

Type	Name	Return Attribute
HeaderVar	OAM_REMOTE_USER	uid

9.4 Provisioning WebGate and Configuring OAM 10*g* (10.1.4.3) and the IAP for IBM WebSphere

This section provides the steps to obtain the OAMCfgTool, provision the required WebGate, create a form authentication scheme, and create a policy domain and OAM 10*g* (10.1.4.3) policies for the IAP and IBM WebSphere.

See Also: "Introduction to the Oracle Access Manager 10g (10.1.4.3) Configuration Tool" on page 9-6

To acquire OAMCfgTool and configure OAM 10*g* (10.1.4.3) for the IAP for IBM WebSphere:

1. Obtain the OAMCfgTool as follows:
   a. Log in to Oracle Technology Network at:

      http://www.oracle.com/technology/software/products/middleware/htdocs/111110_fmw.html

   b. Locate the OAMCfgTool ZIP file with Access Manager Core Components (10.1.4.3.0):

      oamcfgtool<*version*>.zip

   c. Extract and copy oamcfgtool.jar to the computer hosting the IBM WebSphere application to protect.
   d. Confirm that JDK 1.6 (or the latest version) is installed and configured on the host computer.
   e. Change to the file system directory containing OAMCfgTool.
2. Provision WebGate, Create the Authentication Scheme, and Policy Domain: Run the following command using values for your environment. For example:

```
(echo ldappwdjava -jar oamcfgtool.jar
mode=CREATE app_domain=OAMPolicy_for_WAS-IAP
uris_file=/path-to-template-config-file
web_domain=host-id-name
ldap_host=wxyz
ldap_port=6633
ldap_userdn=orcladmin
ldap_base=ldap-base-dn
oam_aaa_host=abcd
oam_aaa_port=7789
oam_aaa_mode=open
log_file=OAMCfg_date.log
log_level=INFO
output_ldif_file=<LDIF_filename>
-noprompt
```

3. Review the information provided by the tool. For example, the parameter and values in Step 3 provide the following information:

```
Processed input parameters
Initialized Global Configuration
Successfully completed the Create operation.
 Operation Summary:
     Policy Domain  : OAMPolicy_for_WAS-IAP
     Host Identifier: OAMPolicy_for_WAS-IAP
     Access Gate ID : OAMPolicy_for_WAS-IAP_AG
```

4. Update host identifiers to include possible host-variations.
5. Add following authorization actions to the "Header Based Assertion" Policy.

Type	Name	Return Attribute
HeaderVar	OAM_REMOTE_USER	uid

6. Proceed to "Installing the Required WebGate for the IHS Web Server" on page 9-11.

9.5 Provisioning and Configuring OAM 11g for the IAP and IBM WebSphere

This section provides the following topics:

- About Provisioning WebGates and AccessGates with OAM 11g
- Provisioning Agents and Creating OAM 11g Policies for IBM WebSphere

9.5.1 About Provisioning WebGates and AccessGates with OAM 11g

This topic introduces OAM 11g access clients, known as policy-enforcement agents, and the process that is required to set up the trust mechanism between the agent and Oracle Access Manager 11g SSO. The process is known as provisioning (also known as registering an agent).

Only registered policy enforcement agents can communicate with an OAM Server, and process information when a user attempts to access a protected resource. Users with valid OAM Administrator credentials can register an OAM Agent using the Administration Console.

You can register a WebGate agent before you install it. Required WebGate or AccessGate configuration files are created during registration and stored in the following path:

DOAMIN_NAME/output/$*Agent_NAME*

During registration, you can also create an application domain and default policies. For this reason, registering an agent is also known as "registering a partner application".

During registration, the Agent is presumed to be on the same Web server as the application it is protecting. However, the Agent can be on a proxy Web server and the application can be on a different host.

During Agent registration:

- One key is generated per agent, accessible to the WebGate through a local wallet file on the client host, and to OAM Server through the Java Key Store on the server side.

  The Agent specific key must be accessible to WebGates through a secure local storage on the client machine.
- A key is generated for the partner (application) during registration. (except for 10*g* (10.1.4.3) WebGate agents).
- An OAM application domain is created, named after the Agent, and populated with default authentication and authorization policies. The new application domain uses the same host identifier that was specified for the Agent during registration.

After registration, agent details appear in the OAM Administration Console and are propagated to all Managed Servers in the cluster. If you choose to automatically create policies during agent registration, you can also view and manage the application domain and policies that were registered with the partner application.

Table 9–3 describes each of named text fields where you enter requested information on the Create OAM Agent page.

Table 9–3 Create OAM Agent Pages for OAM 10g (10.1.4.3) and 11g Agents

OAM Agent Element	Description
Agent Name	The identifying name for this WebGate Agent. This is often the name of the computer that is hosting the Web server used by WebGate. **Note**: If the Agent Name exists, an error occurs and registration fails. If the host identifier exists, the unique Agent Base URL is added to the existing host identifier and registration proceeds.
Agent Base URL Optional	The host and port of the computer on which the Web server for the agent is installed. For example, http://*my_ohs_host:port* or https://*my_host:port*. The port number is optional. **Note**: A particular Agent Base URL can be registered once only. There is a one-to-one mapping from the Agent's Base URL to the Web server domain on which the WebGate is installed (as specified with the <hostidentifier> element). However, one domain can have multiple Agent's Base URLs.
Access Client Password	An optional, unique password for this WebGate, which was assigned during WebGate registration. When a registered WebGate connects to an OAM 11g Server, the password is used for authentication to prevent unauthorized WebGates from connecting to OAM 11g Servers and obtaining policy information.
Security	Level of communication transport security between the Agent and the OAM Server (this must match the level specified for the OAM Server): ■ Open--No transport security ■ Simple--SSL v3/TLS v1.0 secure transport using dynamically generated session keys ■ Cert--SSL v3/TLS v1.0 secure transport using server side x.509 certificates. Choosing this option displays a field where you can enter the Agent Key Password, discussed separately within this table.
Host Identifier	This identifier represents the Web server host.
Auto Create Policies	During agent registration, you can have authentication and authorization policies created automatically. This option is checked (enabled) by default. Default: Enabled **Note**: If you already have a domain and policies registered, you can simply add new resources to it. If you clear this option (no check), no application domain or policies are generated automatically.
Protected Resource (URI) List	URIs for the protected application: /myapp/login, for example. Each URI for the protected application should be specified in a new row of the table for the Protected Resource List. Default: 2 resources are protected by default. /.../* / The default matches any sequence of characters within zero or more intermediate levels spanning multiple directories. Add all IBM WebSphere resources to be protected to this list.
Public Resource (URI) List	Each public application should be specified in a new row of the table for the Public Resource List. Add a field and enter URI values for the public applications and resources. Each URI should be specified in a new row of the table for the Public Resource List. Add all IBM WebSphere resources that should not be protected to this list. Note: /Authen/SSOToken is an additional public resource that is used by the Oracle Access Manager Identity Assertion Provider.

See Also: *Oracle Fusion Middleware Administrator's Guide for Oracle Access Manager with Oracle Security Token Service* for more information

9.5.2 Provisioning Agents and Creating OAM 11g Policies for IBM WebSphere

This topic describes how to provision agents and create policies for OAM 11g.

At least one OAM Server instance must be running in the same mode as the agent. Otherwise, agent registration fails. After provisioning, you can change the communication mode of the OAM Server if needed. Communication between the agent and server continues to work as long as the WebGate mode is at least at the same level as the OAM Server mode (or higher).

To register an agent and create policies for the OAM 11g IAP for IBM WebSphere:

1. Log in to the OAM 11g Administration Console as usual. For example: http://*host:port*/oamconsole.
2. On the Welcome page, click Add OAM 10*g* (10.1.4.3) Agent in the Agent Configuration panel to open a fresh page:

   Alternatively: From the System Configuration tab, expand the Agents node, the OAM Agents node, and the 10*g* (10.1.4.3) Webgates node, then click the Create command button in the tool bar.
3. On the Create: OAM Agent page, enter required details (those with an *) to register this OAM Agent, as shown in Table 9–3.
4. **Protected Resource List**: In this table, enter individual resource URLs to be protected by this OAM Agent, as shown in Table 9–3.
5. **Public Resource List**: In this table, enter individual resource URLs to be public (not protected), as shown in Table 9–3, including /Authen/SSOToken used by the Oracle Access Manager Identity Assertion Provide.
6. Confirm that the Auto Create Policies box is checked (or clear the box to disable this function).
7. Click Apply to submit the registration (or close the page without applying changes).
8. Check the Confirmation window for the location of generated artifacts and then close the window.
9. Repeat steps in this procedure to register an additional AccessGate and policies for use by WebGate and:
   - Enter a name for this registration.
   - Select the appropriate Security mode.
   - Do not specify a Base URL.
   - Check Auto Create Policies
   - Click Apply
10. Proceed to "Installing the Required WebGate for the IHS Web Server".

9.6 Installing the Required WebGate for the IHS Web Server

After provisioning, you can install the OAM 10*g* (10.1.4.3) WebGate for IHS to operate within either an OAM 10*g* (10.1.4.3) or OAM 11g deployment as described here. Ignore any steps that do not apply to your environment.

To download and install the 10*g* (10.1.4.3) WebGate for IHS:

1. Locate and download the WebGate installer as follows:

a. Go to Oracle Fusion Middleware 11gR1 Software Downloads at:

   http://www.oracle.com/technology/software/products/middleware/htdocs/fmw_11_download.html

b. Click **Accept License Agreement**, at the top of the page.

c. From the **Access Manager WebGates (10.1.4.3.0)** row, click the download link for the desired platform and follow on-screen instructions.

d. Store the WebGate installer in the same directory with any 10*g* (10.1.4.3) Access System Language Packs you want to install.

2. Launch the WebGate installer for your platform, installation mode, and Web server, and then:

   a. Dismiss the Welcome screen by clicking Next.

   b. Respond with administrator privileges when asked.

   c. Specify the installation directory for the WebGate. For example:

      /OracleAccessManager/WebComponent/

   d. **Linux or Solaris**: Specify the location of the GCC runtime libraries on this computer.

   e. **Language Pack**—Choose a Default Locale and any other Locales to install, then click Next.

   f. Record the installation directory name in the preparation worksheet if you haven't already, then click Next to continue.

      The WebGate installation begins, which may take a few seconds. On Windows systems, a screen informs you that the Microsoft Managed Interfaces are being configured.

3. **OAM 10*g* (10.1.4.3) Deployment**: Continue installation, as described in the 10*g* (10.1.4.3) *Oracle COREid Access and Identity Installation Guide*, and:

   a. Specify the same values when you install the WebGate that were specified when provisioning the WebGate using OAMCfgTool, earlier.

   b. Specify any additional requested values to properly finish the installation

   c. Copy the files to the WebGate host: *WebGate_install_dir*/access/oblix/config.

   d. Restart the WebGate Web server.

   e. Proceed to "Preparing the IHS Web Server" on page 9-13.

4. **OAM 11g Deployment**: Cancel the WebGate installer (without finishing) and gather WebGate 10*g* (10.1.4.3) provisioning artifacts (and certificate files, if needed). For example:

   a. On the OAM AdminServer host, locate and copy the updated OAM Agent ObAccessClient.xml configuration file (and any certificate artifacts). For example:

      DOMAIN_HOME/output/$*Agent_Name*/

      ObAccessClient.xml
      password.xml (if needed)
      aaa_key.pem (your private key generated by openSSL)
      aaa_cert.pem (signed certificates in PEM format)

b. On the OAM Agent host, add the artifacts to the WebGate directory path. For example:

WebGate_install_dir/access/oblix/lib/ObAccessClient.xml
WebGate_install_dir/access/oblix/config

c. Restart the WebGate Web server.

d. Run the EditHTTPConf tool to update IHS Server configuration for WebGate.

e. Restart the OAM Server that is hosting the Agent.

f. Proceed to "Preparing the IHS Web Server" on page 9-13.

9.7 Preparing the IHS Web Server

When you have 10*g* (10.1.4.3) IHS2 WebGate (or later), the IHS httpd.conf file includes entries for adding the /oamsso directory to the Web Server root. However, if you have an earlier Oracle Access Manager IHS2 WebGate, you must add the following entries under the WebGate block of the httpd.conf file.

To prepare the IHS Web server:

1. On the computer hosting the WebGate, locate IHS httpd.conf file and confirm the following entries exist (if they do not add them):

```
Alias /oamsso "<webage-install-dir>/access/oamsso"
<LocationMatch "/oamsso/*">
Satisfy All
</LocationMatch>
```

2. Proceed with "Preparing the Login Form for WebGate".

9.8 Preparing the Login Form for WebGate

This section describes how to acquire the proper Oracle Access Manager forms for use with the provisioned and installed 10*g* (10.1.4.3) IHS WebGate. No login forms are used from WebGate

If you have OAM 11g, the OAM 11g Server instance provides the Login form and you can skip this procedure.

> **Note:** The forms provided with 10*g* (10.1.4.3) WebGates cannot be used with OAM 11g Servers.

In an OAM 10*g* (10.1.4.3) deployment, if you have:

- 10*g* (10.1.4.3) IHS2 WebGate (or later), find login.html in *WebGate_install_dir*/access/oamsso/login.html.
- Earlier 10*g* (10.1.4.3) IHS2 WebGate, you must create the directory and place a sample login.html file manually, as described in the following procedure.

To preview the login.html file for 10*g* (10.1.4.3) IHS WebGate:

1. OAM 10*g* (10.1.4.3) with 10*g* (10.1.4.3) IHS2 WebGate (or later), preview login.html in *WebGate_install_dir*/access/oamsso/login.html.
2. OAM 10*g* (10.1.4.3) with 10*g* (10.1.4.2.0) or earlier WebGate for IHS2:

a. Create an /oamsso subdirectory in the following path: *WebGate_install_dir*/oamsso.

b. Create and add to the new /oamsso directory a login.html file with the following elements:

```
<!--Sample login Page Code -->
<form name="loginForm" method="post" action="/access/sso">
<b> Username: </b> <input name="userid" type="text" maxLength="80"
size="20" value="">
<b> Password: </b> <input type="password" maxLength="255" size="20"
name="password" autocomplete="off">
<input type="submit" value="Login" name="submit">
</form>
```

3. Proceed to "Configuring IBM WebSphere for OAM SSO and the IAP".

9.9 Configuring IBM WebSphere for OAM SSO and the IAP

This section provides the following topics:

- Configuring a Stand Alone LDAP Registry for OAM in IBM WebSphere
- Adding and Configuring a Virtual Host in IBM WebSphere
- Configuring IHS Reverse Proxy in the IBM WebSphere Console
- Creating the Interceptor Entry in the IBM WebSphere Console
- Configuring the OAM TAI Configuration File

9.9.1 Configuring a Stand Alone LDAP Registry for OAM in IBM WebSphere

This section describes how to configure a stand-alone LDAP registry for OAM within IBM WebSphere.

To configure a stand alone LDAP registry for OAM in IBM WebSphere:

1. Login to your IBM WebSphere console. For example:

   `http://host:port/ibm/console`

2. Go to Security, Global Security.
3. Under User account repository in Available realm definitions, select Standalone Ldap Registry and click Configure.
4. Under General Properties, fill in fields to configure the LDAP directory that is used by OAM:

   Primary administrative user name <OAM admin username>
   Server user identity: keep the default selection
   Type of Ldap Server: <LDAP Directory Type for OAM>
   Host: < host name where LDAP directory resides>
   Port : <LDAP directory bind port>
   Base DN: <LDAP base DN>
   Bind DN: <LDAP bind DN>
   Password: <LDAP password>
   Search timeout: keep the default value (120 seconds)
   Keep default Reuse connection and Ignore case for authorization (checked)

5. Click Apply and OK and save this configuration.

6. On the same page, under Additional Properties, click Advanced Lightweight Directory Access Protocol (LDAP) user registry settings and fill in fields under the General Properties:

   User filter: (&(uid=%v)(objectclass=inetOrgPerson))
   Group filter: (&(cn=%v)(objectclass=ldapsubentry))
   User ID Map: uid
   Group ID Map: cn
   Group Member ID Map: nsRole:nsRole

7. Click Apply and OK and save this configuration.
8. On the same page, under Related Items, click Trusted authentication realms - inbound and confirm that the LDAP entry (host:port) is trusted.
9. Click Test connection to verify the connection configuration.
10. Restart IBM WebSphere.

    If Standalone LDAP Registry is not selected as "Current realm" then under "User account repository" in "Available realm" definitions, select "Standalone Ldap Registry" and click "Set As Current".

11. From now onward, log in to the IBM WebSphere console using OAM LDAP directory login credentials (as registered with IBM WebSphere).

9.9.2 Adding and Configuring a Virtual Host in IBM WebSphere

You must bind your Web applications to virtual hosts (logical name for configuring Web applications to a particular host name). When you request a resource, IBM WebSphere maps the request to an alias of a defined virtual host.

To add and configure a virtual host in IBM WebSphere for the enterprise application:

1. Login to your IBM WebSphere console. For example:
   http://host:port/ibm/console
2. Go to Environment, Virtual Hosts, and click New
3. Enter the General Properties for your environment, as follows:
   a. Add name: *IHS host name* and click on Ok and then save the changes.
   b. Click the recently created entry *IHS host name*:
4. Under Additional Properties, click Host Aliases, and then click New.
5. Fill in details for General Properties for your environment, as follows:
   a. Host: *Host name where IHS server resides*
   b. Port: *IHS port*
6. Click OK to save the changes and continue with the next steps to configure the virtual host in your deployed enterprise application.
7. Go to Applications, WebSphere Enterprise Applications, and:
   a. Click <enterprise application>.
   b. Under Web Module Properties, click Virtual Hosts.
   c. Select all the Web modules and apply the virtual host that you added.
   d. Click OK, then save the changes.
8. Restart IBM WebSphere where the enterprise application is deployed.

9. Proceed to "Configuring IHS Reverse Proxy in the IBM WebSphere Console".

9.9.3 Configuring IHS Reverse Proxy in the IBM WebSphere Console

This section describes how to configures the IHS server in reverse proxy mode within the IBM WebSphere console.

To configure IHS in reverse proxy mode within IBM WebSphere:

1. Login to your IBM WebSphere console. For example:

   ```
   http://host:port/ibm/console
   ```

2. Go to Server Types, Web Servers.
3. Click New, and provide IHS Web server details.
4. Save changes to see a server entry for IHS.
5. Select the *ServerName* and click Generate Plug-in.
6. Select the *ServerName* and click Propagate Plug-in:
7. Configure the IHS Web server to act as a reverse proxy for IBM WebSphere, as follows:
   a. Locate plugin-cfg.xml in *IHS_install_dir*/Plugins/config/*ServerName*
   b. Remove the following entry:

      ```
      <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid"
      Name="/*"/>
      ```

8. Restart the IHS Web server.
9. Proceed to "Creating the Interceptor Entry in the IBM WebSphere Console".

9.9.4 Creating the Interceptor Entry in the IBM WebSphere Console

Tasks are the same whether you are using Oracle Access Manager 10*g* (10.1.4.3) or Oracle Access Manager 11g.

At runtime, the IBM WebSphere extension class loader loads classes. The extension class loader class path is specified by the `ws.ext.dirs` system property. Therefore, you must add the IAP for IBM WebSphere OAMTrustAssociationInterceptor.jar file in the IBM WebSphere classpath:

The IAP for IBM WebSphere OAMTrustAssociationInterceptor.jar file is available from the following path:

```
MW_HOME/oracle_common/modules/oracle.oamprovider_11.1.1/
     OAMTrustAssociationInterceptor.jar
```

To add the OAMTrustAssociationInterceptor.jar to the IBM WebSphere classpath:

1. In IBM WebSphere console go to Servers, Server Types, WebSphere Application, Servers, and select the appropriate server.
2. Under the Server Infrastructure section, click Java And Process Management, and then Process Definition.
3. In Additional properties, select Java Virtual Machine, Custom Properties.
4. In the property `ws.ext.dirs`, add the value for `OAMTrustAssociationInterceptor.jar`. For example:

```
MW_HOME/oracle_common/modules/oracle.oamprovider_11.1.1/
OAMTrustAssociationInterceptor.jar
```

5. Confirm that the two values are separated by colon.
6. Create the Interceptor entry for the OAM IAP, as follows:
   a. In the IBM WebSphere console, go to **Security**, **Global Security**, and ensure that "**Enable Application Security**" is checked.
   b. Under the "**Authentication**" section, click "**Web and SIP Security**" tab, and then click the **Trust association** link.
      a. Under **General Properties**, check the "Enable Trust Association".
      b. Under **Additional Properties**, click Interceptors link.
   c. Under **General Properties**, click **Under New**, and provide the Interceptor class name as follows:

      ```
      oracle.security.was.providers.tai.OAMTrustAssociationInterceptorImpl
      ```

7. Proceed to "Configuring the OAM TAI Configuration File" to configure oamtai.xml as a custom property of Interceptor class path.

9.9.5 Configuring the OAM TAI Configuration File

The oamtai.xml configuration file is used by the OAM Trust Association Interceptor. You must configure the file and modify it for your environment. For details, see:

- About Configuring the OAM TAI Configuration File
- Configuring the OAM TAI Configuration File

9.9.5.1 About Configuring the OAM TAI Configuration File

The oamtai.xml configuration file is available in the following path:

```
MW_HOME/oracle_common/modules/oracle.oamprovider_11.1.1/domain_config_was
/oamtai.xml
```

This file stores the details that are used by the TAI at run time to establish a connection with 10*g* (10.1.4.3) OAM Access Server (or 11g OAM Server).

There are two ways to configure the oamtai.xml file:

- Either copy oamtai.xml to *was_profile_dir*`/config/cells/`*cell_name*`/fmwconfig/oamtai.xml`.
- Or perform Step 1 in the following procedure to configure oamtai.xml as a custom property of the Interceptor entry added earlier.

You must modify the oamtai.xml file to establish a connection to the Access Server, using parameters in Table 9–4 and values for your deployment. To enable Header based assertion, ensure that the Header Based Assertion section in oamtai.xml is not commented and use the same `customHeadername` in both oamtai.xml and the OAM policy.

Table 9–4 oamtai.xml Configuration File Parameters

Parameter	Required or Not	Description
hostPort	Required	Hostname and port of the IHS Web server where the resource is hosted. Note: The host:port should be one of the host name variations present in OAM.
resource	Required	The URL to the protected resource. Default = /Authen/SSOToken or the value in the OAM policy if you have updated it.
ip	Optional	IP address of the client computer that needs to access the resource.
operation	Required	Operation requested to access the Authen/SSOToken.
accessGateName	Required	A unique name, without spaces, that identifies the AccessGate to be used while interacting with OAM. With OAMCfgTool the name is derived from the app_domain value, appended with _AG.
AccessGatePassword	Required	A unique password to verify and identify the AccessGate when interacting with OAM. This prevents unauthorized AccessGates from connecting and obtaining policy information. With OAMCfgTool, this is specified with the app_agent_password parameter. This should differ for each WebGate/AccessGate instance.
accessServerHost	Required	OAM Access Server (or OAM 11g Server) host name.
accessServerPort	Required	OAM Access Server (or OAM 11g Server) port number.
accessServerName	Optional	Name of the OAM Access Server, as identified in the profile (or OAM 11g Server registration).
transportSecurity	Required	The level of transport security between the 10*g* (10.1.4.3) Access Server and associated WebGates must match. The default value is Open. You can specify a different value with OAMCfgTool oam_aaa_mode value. The following parameters trustStore, keyStore, keyStorePass and globalPass values are required when transport security mode is 'Simple' or 'Cert' ■ trustStore: Specify the absolute path to the trust store. ■ keyStore: Specify the absolute path to the key store ■ keyStorePass: Specify the keystore password, ■ globalPass: Specify the global passphrase value that was defined during IHS WebGate installation and configuration.
debug	Required	Turns OAM debugging on or off. Default: false
minConn	Required	The minimum number of connections that this AccessGate can establish with Access Servers. This number must be the same as or less than the number of Access Servers that are actually associated with the WebGate.
maxConn	Required	The maximum number of connections that this AccessGate can establish with Access Servers. This number must be the same as or greater than the number of Access Servers that are actually associated with the WebGate.
timeOutForConnPool	Required	Connection pool time out period. Specify any value in milliseconds. Default: 30000 (milliseconds)

Table 9–4 (Cont.) oamtai.xml Configuration File Parameters

Parameter	Required or Not	Description
Anonymous	Required	Configures the anonymous user value. Note: Following two parameters assertionType and customHeaderName are required for Header Based Assertion. Uncomment it if and only if in case of Header based assertion. ▪ If user configures the headername here, then the same name will be used to configure as return attribute in OAM policy. And don't change the value of assertion type parameter only uncomment parameter entry ▪ If user will not be configuring the header name here, then default header name is "OAM_REMOTE_USER" and same should be configured in OAM policy. Also don't change the value of assertion type parameter only uncomment parameter entry
assertionType	Required	The value should be 'HeaderBasedAssertion', don't change it
customHeaderName	Required	Default value used is " OAM_REMOTE_USER", or according to the OAM Policy if you have updated it. Note: You can provide any value as long as the same value is used in the OAM policy while configuring the Header. Otherwise you must use the default value "OAM_REMOTE_USER" while configuring the policy. In both cases, ensure that the "assertionType" parameter entry in the oamtai.xml file is uncommented.

Note: WebGate timeout should be greater than LTPA timeout. Otherwise, the IAP is not triggered which could cause the WebGate session to time out. If this occurs, a user who logs in with a different userID could get access to the resource because the previously generated LTPA token still exists. LTPA timeout default value is 120 minutes; therefore, the WebGate profile requires a WebGate timeout value greater than 120 minutes.

9.9.5.2 Configuring the OAM TAI Configuration File

The following procedure describes how to configure oamtai.xml for your environment.

Skip Step 1 if oamtai was copied to the following path: *was_profile_dir*/config/cells/*cell_name*/fmwconfig/oamtai.xml.

To configure oamtai.xml as a custom property of the Interceptor:

1. **Custom Interceptor Property**:
   a. In the IBM WebSphere console, go to **Security**, **Global Security**.
   b. Under the "**Authentication**" section, click "**Web and SIP Security tab**"; click the **Trust association** link.
   c. Click the **Trust association** link.
   d. Under **Additional Properties**, click Interceptors link.
   e. Select the Interceptor class name `oracle.security.was.providers.tai.OAMTrustAssociationInterceptorImpl`
   f. Under **Custom Properties**, add a property with the absolute path of oamtai.xml details for the oamtai.xml file:

      Name: *OAMTaiProperty*

      Value: *was_profile_dir*/config/cells/*cell_name*/fmwconfig/oamtai.xml

2. **Modify oamtai.xml**: Use parameters in Table 9–4 with values for your deployment to a establish a connection with the Access Server.
3. **Header Based Assertion**: In the oamtai.xml file, perform the following steps.
   a. Uncomment the "assertionType" entry and retain the value "HeaderBasedAssertion".
   b. Uncomment the "customHeaderName" entry and set the value as desired (Table 9–4).
4. Save the file.
5. **OAM Policy:** Use the same "customHeaderName" value when configuring the OAM policy.
6. Restart IBM WebSphere for changes to take affect.

9.10 Configuring SSO Logout for OAM IAP for IBM WebSphere

This section describes logout with the OAM IAP for IBM WebSphere.

- Configuring Logout for Generic (or Non-ADF) Applications
- Configuring Logout for ADF-Coded Applications

9.10.1 Configuring Logout for Generic (or Non-ADF) Applications

In non-ADF applications, logout is initiated when an application causes the invocation of the logout.html that is configured as the target in the application's logout link.

The logout.html file can be placed at the Web server's doc root, or it can be part of the IBM WebSphere application.

If you are using your own logout.html, you can embed Example 9–3 JavaScript to invoke "delOblixCookie" upon loading the page body. The LTPAToken is deleted by JavaScript; ObSSOCookie is deleted by WebGate.

```
<body onload="delOblixCookie();">
```

Example 9–3 JavaScript to invoke `delOblixCookie`

```
function delCookie(name,path,domain) {
   var today = new Date();
   var deleteDate = new Date(today.getTime() - 48 * 60 * 60 * 1000); // minus 2
   days
   var cookie = name + "="
            + ((path == null) ? "" : "; path=" + path)
            + ((domain == null) ? "" : "; domain=" + domain)
            + "; expires=" + deleteDate;
   document.cookie = cookie;
}
function delOblixCookie() {
         // set focus to ok button
      var isNetscape = (document.layers);
  if (isNetscape == false || navigator.appVersion.charAt(0) >= 5) {
    for (var i=0; i<document.links.length; i++) {
      if (document.links[i].href == "javascript:top.close()") {
          document.links[i].focus();
          break;
      }
    }
```

```
   }
  delCookie('ObTEMC', '/');
  delCookie('ObSSOCookie', '/');
  delCookie('LtpaToken', '/');
  delCookie('LtpaToken2', '/');
  // in case cookieDomain is configured
  // delete same cookie to all of subdomain
  var subdomain;
  var domain = new String(document.domain);
  var index = domain.indexOf(".");
  while (index > 0) {
     subdomain = domain.substring(index, domain.length);
     if (subdomain.indexOf(".", 1) > 0) {
         delCookie('ObTEMC', '/', subdomain);
         delCookie('ObSSOCookie', '/', subdomain);
         delCookie('LtpaToken', '/', subdomain);
         delCookie('LtpaToken2', '/', subdomain);
     }
     domain = subdomain;
     index = domain.indexOf(".", 1);
   }
}
```

To configure logout for generic (non-ADF) applications:

1. Locate the desired logout.html file.
2. Add the JavaScript in Example 9–3 to logout.html to invoke "delOblixCookie" upon loading the page body.
3. In the Oracle Access Manager policy, protect logout.html using the Anonymous Authentication Scheme, as described in the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Manager with Oracle Security Token Service*.

9.10.2 Configuring Logout for ADF-Coded Applications

In ADF coded Fusion Middleware applications such as Oracle WebCenter Portal application, single sign off is achieved through OPSS. For details, see the following topics:

- Configuring WebGate for Logout
- Configuring OPSS for SSO Logout with Oracle Access Manager
- Configuring oamAuthenProvider.jar in the IBM WebSphere classpath
- Verifying SSO Logout

9.10.2.1 Configuring WebGate for Logout

This topic provides an example (Example 9–4) and procedure that you can use and customize to logout an application protected by OAM 10g with a 10g WebGate

> **Note:** Example 9–4 applies only for an end URI of a single word. For a long URI, you must update the parsing logic accordingly.

To configure WebGate for logout:

1. Create and edit logout.html for the WebGate based on Example 9–4: add and call the function `handleLogout()` for redirecting the logout request to the end URL specified in the logout URL

Example 9–4 Sample logout.html Script

```
<html>
<head>
<script language="javascript" type="text/javascript">

function handleLogout() {

    //get protocol used at the server (http/https)
    var webServerProtocol = window.location.protocol;
    //get server host:port
    var webServerHostPort = window.location.host;
    //get query string present in this URL
    var origQueryString = window.location.search.substring(1);

    //vars to parse the querystring
    var params = new Array();
    var par = new Array();
    var val;

    if (origQueryString != null && origQueryString != "") {

        params = origQueryString.split("&");

        //search for end_url and redirect the user to this
        for (var i=0; i<params.length; i++) {

        par = params[i].split("=");
        if ("end_url" == par[0]) {
          endUrlVal = par[1];

        //check if val (value of end_url) begins with "/" or "%2F" (is it an URI?)
        if (endUrlVal.substring(0,1) == "/" || endUrlVal.substring(0,1) == "%") {
          if (endUrlVal.substring(0,1) == "%")
            endUrlVal = "/" + endUrlVal.substring(3);

         //modify the end_url value now
           endUrlVal = webServerProtocol + "//" + webServerHostPort + endUrlVal;
         }
    //redirect the user to this URL
    window.location.href = endUrlVal;
         }
       }
   }
}
</script>
</head>
<body onLoad="handleLogout();">
<h3>You have been logged out<h3>

</body>
</html>
```

2. Store your logout.html script to *WebGate_install_dir*/oamsso/logout.html
3. In the httpd.conf file, ensure following entries exist under the WebGate block:

```
Alias /oamsso "<webage-install-dir>/access/oamsso
<LocationMatch "/oamsso/*">
Satisfy All
</LocationMatch>
```

4. Proceed to "Configuring OPSS for SSO Logout with Oracle Access Manager".

9.10.2.2 Configuring OPSS for SSO Logout with Oracle Access Manager

Application configuration for logout depends on whether you have an ADF-coded application integrated with OPSS versus not integrated with OPSS. This topic focuses on ADF-coded applications that are integrated with OPSS.

The following procedure is similar to configuring logout for 10g WebGates, with a specific step for ADF-coded applications, which must send the end_url value to identify where to redirect the user after logout processing. However, with ADF-coded applications, logout occurs when the application causes the following URI to be invoked:

```
/<app context root>/adfAuthentication?logout=true&end_url=<any uri>
```

To configure OPSS for SSO Logout with OAM:

1. Locate and open the jps-config .xml file in the following path:

   was_profile_dir/config/cells/*cell_name*/fmwconfig/jps-config.xml

2. Within jps-config .xml, add the following **<propertySet name="props.auth.uri.0">** element and values:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<jpsConfig xmlns="http://xmlns.oracle.com/oracleas/schema/11/jps-config-11_
1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/oracleas/schema/11/jps-config-11_
1.xsd">
<property value="off" name="oracle.security.jps.jaas.mode"/>
<propertySets>
.
<propertySet name="props.auth.uri.0">
<property value="/oamsso/logout.html" name="logout.url"/>
<property value="${app.context}/adfAuthentication" name="login.url.BASIC"/>
<propertyvalue="${app.context}/adfAuthentication"name="login.url.ANONYMOUS"/>
<property value="${app.context}/adfAuthentication" name="login.url.FORM"/>
</propertySet>
<propertySet name="props.auth.level.0">
<property value="0" name="type-level:ANONYMOUS"/>
<property value="1" name="type-level:BASIC"/>
<property value="2" name="type-level:FORM"/>
.
</propertySets>
```

3. Within jps-config .xml, add the following **<serviceProviders>** element and values:

```
...
</propertySets>
<serviceProviders>
<serviceProvider class="oracle.security.jps.internal.sso.SsoService
Provider" name="sso.provider.0" type="SSO"/>
    </serviceProviders>
```

4. Within jps-config .xml, add the following **<serviceInstances>** element and values:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
...
</serviceProviders>
<serviceInstances>
.
.
<serviceInstance provider="sso.provider.0" name="sso.inst.0">
<property value="oracle.security.jps.wls.internal.sso.WlsToken
Provider" name="token.provider.class"/>
<property value="2" name="default.auth.level"/>
<property value="oracle.security.wls.oam.providers.sso.OAMSSO
ServiceProviderImpl" name="sso.provider.class"/>
<property value="OAMSSOToken" name="token.type"/>
<propertySetRef ref="props.auth.uri.0"/>
<propertySetRef ref="props.auth.level.0"/>
</serviceInstance>
.
.
</serviceInstances>
```

5. Within jpsContexts, add the highlighted <serviceInstanceRef ref="sso.inst.0"/> element and value:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
...
</serviceInstances>
<jpsContexts default="default">
<jpsContext name="default">
<serviceInstanceRef ref="credstore"/>
<serviceInstanceRef ref="keystore"/>
<serviceInstanceRef ref="policystore.xml"/>
<serviceInstanceRef ref="audit"/>
<serviceInstanceRef ref="idstore.ldap"/>
<serviceInstanceRef ref="sso.inst.0"/>
</jpsContext>
</jpsContexts>
</jpsConfig>
```

6. In the Oracle Access Manager policy, protect `/oamsso/logout.html` with the Anonymous Authentication scheme, as described in theOracle Fusion Middleware Administrator's Guide for Oracle Access Manager with Oracle Security Token Service.
7. Proceed to "Configuring oamAuthenProvider.jar in the IBM WebSphere classpath".

9.10.2.3 Configuring oamAuthenProvider.jar in the IBM WebSphere classpath

To perform logout through OPSS, you must configure oamAuthnProvider.jar in the IBM WebSphere classpath. This is similar to adding the interceptor jar in the IBM WebSphere classpath in "Creating the Interceptor Entry in the IBM WebSphere Console" on page 9-16.

The oamAuthnProvider.jar file is available from the following path:

```
MW_HOME/oracle_common/modules/oracle.oamprovider_11.1.1/
    oamAuthnProvider.jar
```

To add oamAuthnProvider.jar to the IBM WebSphere classpath:

1. In the IBM WebSphere console go to Servers, Server Types, WebSphere Application, Servers, and select the appropriate server.

2. Under the Server Infrastructure section, click Java And Process Management, and then click Process Definition.
3. In Additional properties, select Java Virtual Machine, Custom Properties.
4. In the ws.ext.dirs property, add the value for oamAuthnProvider.jar after the entry for OAMTrustAssociationInterceptor.jar and confirm that the two values are separated by a colon. For example:

```
ws.ext.dir   MW_HOME/oracle_common/modules/oracle.oamprovider_11.1.1/
   OAMTrustAssociationInterceptor.jar:MW_HOME/oracle_common/modules/
   oracle.oamprovider_11.1.1/oamAuthnProvider.jar
```

5. Restart IBM WebSphere.
6. Proceed to "Verifying SSO Logout"

9.10.2.4 Verifying SSO Logout

To verify SSO logout:

1. From a browser, enter the URL of the protected resource. For example:

```
http://host:port/<app context root>/adfAuthentication
```

2. Confirm that the login page appears and sign in using proper credentials
3. Confirm that the protected resource is served
4. Open a new browser tab or window and access the same resource to confirm that the second attempt does not require another login
5. Logout from one tab using a URL like the following sample:

```
http://host:port/<app context root>/adfAuthentication?logout=true&end_url=<any
uri>
```

6. Access the resource again to confirm that a login page appears.

9.11 Known Issues

Problem:

Oracle Access Manager Identity Assertion Provider for IBM WebSphere does not support the Simple security mode.

Problem: Inconsistent

Oracle Access Manager Identity Assertion Provider for IBM WebSphere does not generate an LTPA token after successful authentication and valid ObSSOCookie generation.

```
Error
403: AuthenticationFailed
```

And the following error in the trace log:

```
com.ibm.websphere.security.WebTrustAssociationFailedException: Can not assert
user identity as LoggedIn user value is null
```

Solution

Refresh the browser 2 or 3 times. A valid LTPA token is generated.

For the server to communicate with a client in Simple transport security mode, a Master Secret Key is required. Sun JDK has an API that generates the Master Secret Key. However, IBM WebSphere contains the IBM JDK which does not have the API to generate the Master Secret Key.

A

Fusion Middleware Control Page Reference

This appendix describes the features and options available on the Fusion Middleware Control pages that appear when you are managing an IBM WebSphere cell that was configured for Oracle Fusion Middleware.

This appendix contains the following sections:

- Understanding the Information on the IBM WebSphere Cell Home Page
- Understanding the Information on the WebSphere Application Server Home Page
- Understanding the Information on the IBM WebSphere Application Deployment Home Page

A.1 Understanding the Information on the IBM WebSphere Cell Home Page

The Cell home page is divided into the following regions:

- Summary Region of the Cell Home Page
- Deployments Region of the Cell Home Page
- Servers Region of the Cell Home Page
- Clusters Region of the Cell Home Page

Summary Region of the Cell Home Page

The Summary region of the Cell home page provides general information about the cell, as well as a link to the IBM WebSphere Administrative Console, which you can use to manage the cell.

Table A–1 describes the fields available in the General section of the Summary region.

Table A–1 Fields Available in the General Section of the Summary Region

Element	Description
Cell Name	The name given to the cell when the cell was configured with the Oracle Fusion Middleware Configuration Wizard.
Version	The version of IBM WebSphere that was used to configure the Cell. Note that this version number can also identify which set of patches have been applied to the IBM WebSphere installation.

Table A–1 (Cont.) Fields Available in the General Section of the Summary Region

Element	Description
Administrative Console Port	The non-secure port used to access the IBM WebSphere Administrative Console. Specifically, this is the port identified by *WC_Adminhost_port* in the following URL: `http://hostname:WC_Adminhost_port/ibm/console`
Administrative Console Secure Port	The secure port used to access the IBM WebSphere Administrative Console. Specifically, this is the port identified by *WC_Adminhost_secure port* in the following URL: `https://hostname:WC_Adminhost_secure_port/ibm/console`
SOAP Connector Port	The port used for communications with the administrative server via the Simple Object Access Protocol (SOAP).
Bootstrap Port	This is the value of the bootstrap port for the administrative server. This port is required when you are installing the IBM WebSphere Application Client software and when using utilities such as the IBM WebSphere `dumpNameSpace` tool.
Deployment Mode	The deployment mode of the IBM WebSphere software. For example, this field indicates whether this is an IBM WebSphere Application Server - Network Deployment installation or an IBM WebSphere Application Server deployment.

Deployments Region of the Cell Home Page

This region lists the applications that have been deployed to the servers in the cell. Each application deployment is listed, as well as the deployment name, status, and target servers where the deployment is running.

Click the name of an application deployment to display the WebSphere Application Deployment home page, which provides more information about each application deployment.

The chart identifies the percentage of deployments that are currently up and running, as opposed to those that are down or not available.

Internal applications are those that are required by Oracle Fusion Middleware. The internal applications are deployed automatically and are required by the Oracle Fusion Middleware products you installed and configured in the cell.

Servers Region of the Cell Home Page

This region lists the servers in the cell. The chart identifies the percentage of servers that are up and running, as opposed to those that are down or not available.

For each server, the region lists the server name, status, and--if it resides in a cluster--the name of the cluster.

Clusters Region of the Cell Home Page

This region lists the clusters currently configured in the cell. For each cluster, it provides the cluster name, status, and a list of the servers in the cluster.

A.2 Understanding the Information on the WebSphere Application Server Home Page

The WebSphere Application Server home page is divided into the following regions:

Summary Region of the WebSphere Application Server Home Page

The Summary region of the WebSphere Application Server home page provides general information about the server, as well as a link to the IBM WebSphere Administrative Console, which you can use to manage the server.

Table A–2 describes the fields available in the General section of the Summary region.

Table A–2 Fields Available in the General Section of the Summary Region of the Applicatin Server Page

Element	Description
Cell Name	The name given to the cell when the cell was configured with the Oracle Fusion Middleware Configuration Wizard.
Node Name	The name of the node that contains this server.
Version	The version of IBM WebSphere that was used to configure the Cell. Note that this version number can also identify which set of patches have been applied to the IBM WebSphere installation.
WebSphere Home	The full path of the directory where the current IBM WebSphere software was installed and configured.
Host	The fully-qualified name of the host where the server is currently running.

Deployments Region of the WebSphere Application Server Home Page

This region lists the applications that have been deployed to the server. Each application deployment is listed, including the deployment name and status.

Click the name of an application deployment to display the WebSphere Application Deployment home page, which provides more information about each application deployment.

The chart identifies the percentage of deployments that are currently up and running, as opposed to those that are down or not available.

Internal applications are those that are required by Oracle Fusion Middleware. The internal applications are deployed automatically and are required by the Oracle Fusion Middleware products you installed and configured in the cell.

A.3 Understanding the Information on the IBM WebSphere Application Deployment Home Page

The Application Deployment page is divided into the following sections:

Summary Region on the IBM WebSphere Application Deployment Page

The Summary region of the WebSphere Application Deployment home page provides general information about the application, as well as a link to the IBM WebSphere Administrative Console, which you can use to manage the application.

Table A–3 describes the fields available in the General section of the Summary region.

Table A–3 Fields Available in the General Section of the Summary Region of the Application Deployment Page

Element	Description
Application Type	The type of application. For example, this field indicates whether the application was deployed as an enterprise archive (EAR) or other archive type.
Cell Name	The name given to the cell when the cell was configured with the Oracle Fusion Middleware Configuration Wizard.
Node Name	The name of the node that contains the server where the application was deployed.
Deployed On	The name of the server where this instance of the application is deployed.