Oracle® Fusion Middleware

Third-Party Application Server Guide

11*g* Release 1 (11.1.1)

E17852-01

January 2011

Oracle Fusion Middleware Third-Party Application Server Guide, 11*g* Release 1 (11.1.1)

E17852-01

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, Dan MacKinnon, Mark Miller, Michael Rubino, Reza Shafii, Sitaraman Swaminathan, Ken Vincent

Contents

Preface ix

Audience ix
Documentation Accessibility ix
Related Documents x
Conventions x

1 Introduction to Third-Party Application Servers

1.1 What Is a Third-Party Application Server? 1-1
1.2 Oracle Fusion Middleware Components That Support Third-Party Application Servers 1-1
1.3 Overview of the Oracle Fusion Middleware IBM WebSphere Support 1-1
1.3.1 Supported IBM WebSphere Application Servers 1-2
1.3.2 Understanding the Topology of Oracle Fusion Middleware on IBM WebSphere..... 1-2
1.3.2.1 Typical Oracle Fusion Middleware Topology on IBM WebSphere Application Server - ND 1-2
1.3.2.2 Typical Oracle Fusion Middleware Topology on IBM WebSphere Application Server 1-3
1.4 Documentation Resources When Using Oracle Fusion Middleware on IBM WebSphere 1-3

2 Installing and Configuring Oracle Fusion Middleware on IBM WebSphere

2.1 Task 1: Review the System Requirements and Certification Information 2-1
2.2 Task 2: Obtain the Necessary Software Media or Downloads 2-2
2.3 Task 3: Identify a Database and Install the Required Database Schemas 2-2
2.4 Task 4: Install the IBM WebSphere Software 2-3
2.4.1 IBM Online Resources for Obtaining and Installing the IBM WebSphere Software . 2-3
2.4.2 Important Considerations When Installing the IBM WebSphere Software 2-3
2.5 Task 5: Install Oracle Fusion Middleware 2-3
2.5.1 General Installation Instructions for the Supported Oracle Fusion Middleware Products 2-4
2.5.2 Special Instructions When Installing Oracle Fusion Middleware with IBM WebSphere . 2-4
2.6 Task 6: Configure an LDAP Server for Oracle SOA Suite 2-5
2.7 Task 7: Configure Your Oracle Fusion Middleware Components in a New IBM WebSphere Cell 2-5
2.8 Task 8: Start the IBM WebSphere Servers 2-6

2.9 Task 9: Verify the Installation 2-8

3 Managing Oracle Fusion Middleware on IBM WebSphere

3.1 Summary of the Oracle Fusion Middleware Management Tools on IBM WebSphere 3-1
3.1.1 Using the WebSphere Administrative Console.......... 3-1
3.1.1.1 About the IBM WebSphere Administrative Console 3-1
3.1.1.2 Locating the Port Number and URL of the IBM WebSphere Administrative Console 3-2
3.1.2 Using Oracle Enterprise Manager Fusion Middleware Control.......... 3-2
3.1.2.1 About Oracle Enterprise Manager Fusion Middleware Control.......... 3-2
3.1.2.2 Locating Port Number and URL for Fusion Middleware Control.......... 3-3
3.1.2.3 Displaying Fusion Middleware Control 3-3
3.1.2.4 Viewing an IBM WebSphere Cell from Fusion Middleware Control.......... 3-3
3.1.2.5 Viewing an IBM WebSphere Server from Fusion Middleware Control 3-4
3.1.2.6 Viewing an IBM WebSphere Application Deployment from Fusion Middleware Control 3-4
3.1.2.7 Performing Oracle Fusion Middleware-Specific Administration Tasks for the Cell . 3-5
3.1.2.8 Differences When Using Fusion Middleware Control on IBM WebSphere 3-5
3.1.3 Using the Oracle Fusion Middleware wsadmin Commands.......... 3-7
3.1.3.1 About the Oracle Fusion Middleware wsadmin Command-Line Shell.......... 3-7
3.1.3.2 Starting the Oracle Fusion Middleware wsadmin Command-Line Shell and Connecting to the Deployment Manager 3-7
3.1.3.3 Using the Oracle Fusion Middleware wsadmin Command-Line Online Help .. 3-9
3.1.3.3.1 Listing the Oracle Fusion Middleware wsadmin Command Categories 3-9
3.1.3.3.2 Listing the Commands within an Oracle Fusion Middleware wsadmin Command-Line Category 3-9
3.1.3.3.3 Getting Help on a Specific Oracle Fusion Middleware wsadmin Command...... 3-10
3.1.3.4 Differences Between the wsadmin Commands and the WebLogic Scripting Tool (WLST) Commands 3-11
3.1.3.5 Differences Between Oracle Fusion Middleware wsadmin Commands and IBM WebSphere Wsadmin Commands 3-12
3.2 Basic Administration Tasks on IBM WebSphere.......... 3-12
3.2.1 Starting and Stopping Servers on IBM WebSphere.......... 3-12
3.2.1.1 Starting and Stopping IBM WebSphere Servers with Profile Scripts 3-12
3.2.1.2 Starting and Stopping IBM WebSphere Servers with Fusion Middleware Control... 3-13
3.2.2 Configuring Metadata Services (MDS) on IBM WebSphere 3-13
3.2.2.1 Differences in MDS Command-Line Features on IBM WebSphere.......... 3-13
3.2.2.2 Differences in MDS Fusion Middleware Control Pages on IBM WebSphere .. 3-14
3.2.3 Configuring Oracle Fusion Middleware Logging on IBM WebSphere.......... 3-14
3.2.4 Setting Up the Diagnostic Framework 3-16
3.2.5 Creating a Data Source in an IBM WebSphere Cell.......... 3-16
3.3 Deploying Applications on IBM WebSphere.......... 3-20
3.3.1 Preparing to Deploy Oracle Fusion Middleware Applications on IBM WebSphere......... 3-20
3.3.2 Methods for Deploying Oracle Fusion Middleware Applications on IBM WebSphere ... 3-20

3.3.3 Deploying Applications that Require MDS Deployment Plan Customizations on IBM WebSphere 3-21

4 Managing Oracle SOA Suite on IBM WebSphere

4.1 Configuring Oracle SOA Suite Users and Groups in an External LDAP Server on IBM WebSphere 4-1
4.2 Differences and Restrictions When Developing and Deploying Oracle SOA Suite Applications on IBM WebSphere 4-3
4.2.1 Oracle SOA Suite wsadmin and WLST Command Differences 4-4
4.2.2 Configuring the WebSphere Application Client for Use with Oracle JDeveloper..... 4-5
4.2.2.1 Installing the WebSphere Application Client... 4-5
4.2.2.2 Creating the wsadmin.sh/bat File 4-5
4.2.2.3 Running wsadmin.sh or wsadmin.bat from the Command Line 4-7
4.2.2.4 Editing the sas.client.props File.. 4-7
4.2.2.5 Creating an Application Server Connection in Oracle JDeveloper...................... 4-7
4.2.3 Configuring the Proxy on IBM WebSphere Server... 4-7
4.2.4 Creating an Application Server Connection... 4-8
4.2.5 Deploying SOA Composite Applications 4-11
4.2.6 Using EJB Bindings... 4-12
4.2.6.1 EJB Service Binding 4-12
4.2.6.2 EJB Client 4-13
4.2.6.3 EJB Reference Binding 4-13
4.2.7 AQ Technology Adapter and WebSphere 7.0 4-13
4.2.8 JMS Technology Adapter on WebSphere 7.0.. 4-15
4.2.9 Oracle Database Adapter on WebSphere 7.0.. 4-15
4.3 Differences and Restrictions When Managing Oracle SOA Suite Components on IBM WebSphere 4-15
4.3.1 Publishing Services to a UDDI Registry.. 4-15
4.3.2 Oracle Enterprise Manager Fusion Middleware Control Console Shortcut Links . 4-16

5 Managing Web Services on IBM WebSphere

5.1 Configuring a Default Administrative User from the LDAP Directory 5-1
5.2 Configuring Oracle WSM on IBM WebSphere.. 5-2
5.2.1 Configuring Oracle WSM 5-2
5.2.2 Connecting to the Oracle WSM Policy Manager... 5-3
5.3 Differences and Restrictions When Developing Web Services Applications on IBM WebSphere 5-5
5.3.1 High Availability 5-5
5.3.2 Asynchronous Web Services... 5-6
5.3.3 JDeveloper 5-6
5.4 Differences and Restrictions When Managing Web Services Components on IBM WebSphere 5-6
5.4.1 Automatic Discovery of Oracle WSM Policy Manager... 5-6
5.4.2 Web Services Atomic Transactions 5-6
5.4.3 No Support for Native Web Services.. 5-7
5.4.4 Reliable Messaging... 5-7
5.4.5 Enterprise Manager Fusion Middleware Control.. 5-7

5.5 Using the Web Services wsadmin Commands 5-8
5.5.1 Executing the Web Services wsadmin Commands 5-8
5.5.2 WebServices wsadmin Commands 5-9
5.5.3 wsmManage wsadmin Commands 5-11

6 Managing Oracle Fusion Middleware Security on IBM WebSphere

6.1 IBM WebSphere Identity Stores 6-1
6.1.1 Configuring a Registry 6-2
6.1.2 Seeding a Registry 6-3
6.2 Migrating Policies at Deployment 6-3
6.2.1 jps.policystore.migration 6-4
6.2.2 jps.policystore.applicationid 6-4
6.2.3 jps.policystore.removal 6-4
6.3 Migrating Credentials at Deployment 6-5
6.3.1 jps.credstore.migration 6-5
6.4 Reassociating Policies with reassociateSecurityStore 6-5
6.5 Deployment Mode 6-5
6.6 Configuring the JpsFilter and the JpsInterceptor 6-5
6.7 Using System Variables in Code Source URLs 6-5
6.8 Sample opss-application File 6-6
6.9 Executing Common Audit Framework wsadmin Commands 6-6

7 Managing OAM Identity Assertion on IBM WebSphere

7.1 Introduction to OAM Identity Assertion on IBM WebSphere 7-1
7.1.1 Scenario 1: Oracle Access Manager 10*g* (10.1.4.3) with the IAP on IBM WebSphere 7-2
7.1.2 Scenario 2: OAM 11g with the IAP and IBM WebSphere 7-3
7.2 Installing Components for the Oracle Access Manager IAP for IBM WebSphere 7-5
7.3 Introduction to the Oracle Access Manager 10*g* (10.1.4.3) Configuration Tool 7-6
7.4 Provisioning WebGate and Configuring OAM 10*g* (10.1.4.3) and the IAP for IBM WebSphere 7-7
7.5 Provisioning and Configuring OAM 11g for the IAP and IBM WebSphere 7-9
7.5.1 About Provisioning WebGates and AccessGates with OAM 11g 7-9
7.5.2 Provisioning Agents and Creating OAM 11g Policies for IBM WebSphere 7-11
7.6 Installing the Required WebGate for the IHS Web Server 7-11
7.7 Preparing the IHS Web Server 7-13
7.8 Preparing the Login Form for WebGate 7-13
7.9 Configuring IBM WebSphere for OAM SSO and the IAP 7-14
7.9.1 Configuring a Stand Alone LDAP Registry for OAM in IBM WebSphere 7-14
7.9.2 Adding and Configuring a Virtual Host in IBM WebSphere 7-15
7.9.3 Configuring IHS Reverse Proxy in the IBM WebSphere Console 7-16
7.9.4 Creating the Interceptor Entry in the IBM WebSphere Console 7-16
7.9.5 Configuring the OAM TAI Configuration File 7-17
7.9.5.1 About Configuring the OAM TAI Configuration File 7-17
7.9.5.2 Configuring the OAM TAI Configuration File 7-19
7.10 Configuring SSO Logout for OAM IAP for IBM WebSphere 7-20
7.10.1 Configuring Logout for Generic (or Non-ADF) Applications 7-20
7.10.2 Configuring Logout for ADF-Coded Applications 7-21

7.10.2.1 Configuring WebGate for Logout 7-21
7.10.2.2 Configuring OPSS for SSO Logout with Oracle Access Manager 7-23
7.10.2.3 Configuring oamAuthenProvider.jar in the IBM WebSphere classpath 7-24
7.10.2.4 Verifying SSO Logout 7-25
7.11 Known Issues 7-25

A Fusion Middleware Control Page Reference

A.1 Understanding the Information on the IBM WebSphere Cell Home Page A-1
A.2 Understanding the Information on the WebSphere Application Server Home Page A-2
A.3 Understanding the Information on the IBM WebSphere Application Deployment Home Page A-3

Preface

This preface contains the following sections:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

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

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Access to Oracle Support

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

Related Documents

For more information, see the following 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 Fusion Developer'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:

- What Is a Third-Party Application Server?
- Oracle Fusion Middleware Components That Support Third-Party Application Servers
- Overview of the Oracle Fusion Middleware IBM WebSphere Support
- Documentation Resources When Using Oracle Fusion Middleware on IBM WebSphere

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 products on supported third-party application servers:

- Oracle SOA Suite
- Oracle Application Developer Runtime

For Oracle Fusion Middleware 11*g* (11.1.1.4.0), Oracle supports only IBM WebSphere 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) 7.0.11
- IBM WebSphere Application Server 7.0.11

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 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".

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 When Using Oracle Fusion Middleware on IBM WebSphere

In addition to this document, 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:

- Task 1: Review the System Requirements and Certification Information
- Task 2: Obtain the Necessary Software Media or Downloads
- Task 3: Identify a Database and Install the Required Database Schemas
- Task 4: Install the IBM WebSphere Software
- Task 5: Install Oracle Fusion Middleware
- Task 6: Configure an LDAP Server for Oracle SOA Suite
- Task 7: Configure Your Oracle Fusion Middleware Components in a New IBM WebSphere Cell
- Task 8: Start the IBM WebSphere Servers
- Task 9: Verify the Installation

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* (11.1.1.4.0) or later
- One of the following Oracle Fusion Middleware software products, which are supported on IBM WebSphere:
  - Oracle SOA Suite 11*g* (11.1.1.4.0) or later
  - Oracle Application Development Runtime 11*g* (11.1.1.4.0) or later

  > **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 "Obtain the Oracle Fusion Middleware Software" in the *Oracle Fusion Middleware Installation Planning Guide*.

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

Some Oracle Fusion Middleware products, such as Oracle SOA Suite, require a metadata repository. 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*.

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 When 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 When 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 When Installing the IBM WebSphere Software

When you perform the installation, note the following requirements for Oracle Fusion Middleware products:

- 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.

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

  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*

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:

  ```
  WAS_APPSERVER_HOME/java
  ```

  For example, if you are using the default location for a typical IBM WebSphere application server directory:

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

- 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 with Oracle WebLogic Server, you create the Middleware home before you install the Oracle Fusion Middleware software, 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 with 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 for Oracle SOA Suite

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, "Configuring Oracle SOA Suite Users and Groups in an External LDAP Server on IBM WebSphere".

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

To configure Oracle Fusion Middleware in a new 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.

- 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 pre-configuration 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:

   On Linux systems:

   ```
   MW_HOME/ORACLE_HOME/common/bin/was_config.sh
   ```

   On Windows systems:

   ```
   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 above 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 Linux system:

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

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

   For more information, see "Creating a New Cell" in *Oracle Fusion Middleware Configuration Guide for IBM WebSphere Application Server*.

   Note the following 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 "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.
   - 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*.

2.8 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.

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:

   On UNIX operating systems:

   ```
   profiles/deployment_mgr_name/bin/startManager.sh -profileName dmgr_profileName
   ```

   On Windows operating systems:

```
profiles\deployment_mgr_name\bin\startManager.cmd -profileName dmgr_profileName
```

For example, on a UNIX operating system:

```
/disk01/IBM/WebSphere/ApplicationServer/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:

   On UNIX operating systems:

   ```
   profiles/profile_name/bin/startNode.sh -profileName profileName
   ```

   On Windows operating systems:

   ```
   profiles\profile_name\bin\startNode.cmd -profileName profileName
   ```

   For example, on a UNIX operating system:

   ```
   /disk01/IBM/WebSphere/ApplicationServer/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:

   On UNIX operating systems:

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

   On Windows operating systems:

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

   For example, on a UNIX operating system:

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

4. If you have configured Oracle SOA Suite on IBM WebSphere, then start the Oracle SOA Suite server:

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

   On UNIX operating systems:

   ```
   profiles/profile_name/bin/startServer.sh soa_server1
            -profileName profileName
   ```

   On Windows operating systems:

   ```
   profiles\profile_name\bin\startServer.cmd soa_server1
            -profileName profileName
   ```

   For example, on a UNIX operating system:

   ```
   /disk01/IBM/WebSphere/ApplicationServer/profiles
           /Custom01/bin/startServer.sh soa_server1
   ```

```
                        -profileName Custom01
```

2.9 Task 9: Verify the Installation

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

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

After you install and configure Oracle Fusion Middleware with 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, 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 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 Port Number and URL for Fusion Middleware Control

To locate the port number of the 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 to display Fusion Middleware Control:

```
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.

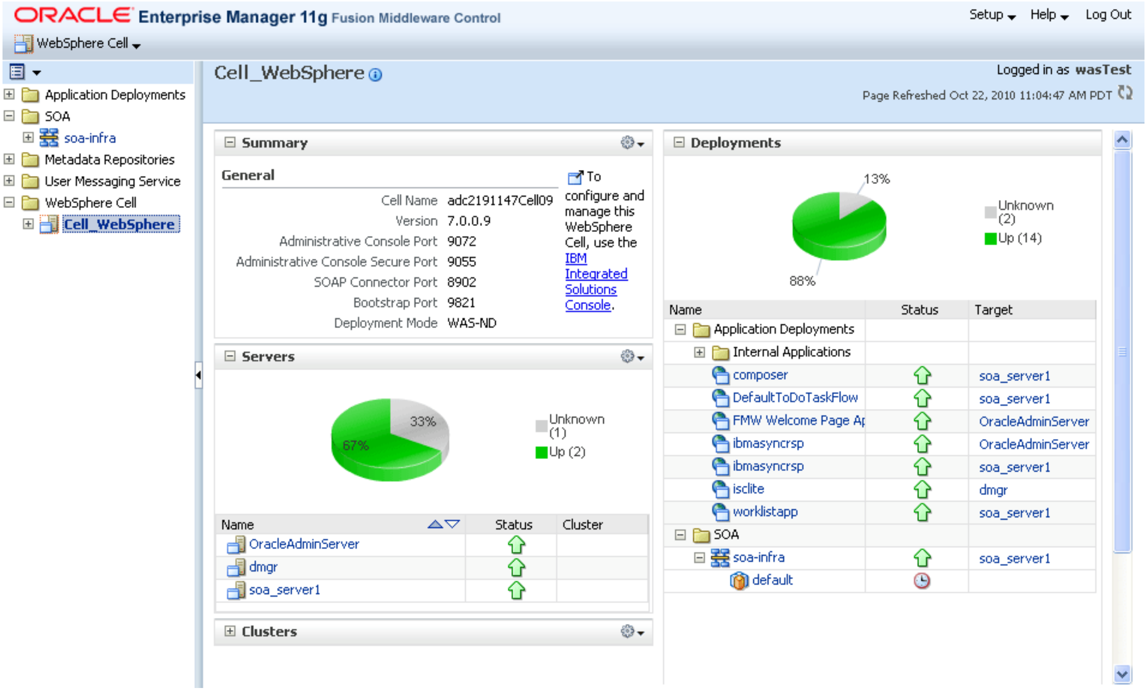
When you first log in to Fusion Middleware Control, the IBM WebSphere Cell home page appears (Figure 3–1). From this page, you can view the servers, applications, and clusters that are associated with the cell.

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



General

Cell Name	adc2191147Cell09
Version	7.0.0.9
Administrative Console Port	9072
Administrative Console Secure Port	9055
SOAP Connector Port	8902
Bootstrap Port	9821
Deployment Mode	WAS-ND

To configure and manage this WebSphere Cell, use the IBM Integrated Solutions Console.

Name	Status	Cluster
OracleAdminServer	↑	
dmgr	↑	
soa_server1	↑	

Name	Status	Target
Application Deployments		
Internal Applications		
composer	↑	soa_server1
DefaultToDoTaskFlow	↑	soa_server1
FMW Welcome Page Ap	↑	OracleAdminServer
ibmasyncrsp	↑	OracleAdminServer
ibmasyncrsp	↑	soa_server1
isclite	↑	dmgr
worklistapp	↑	soa_server1
SOA		
soa-infra	↑	soa_server1
default	🕒	

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 will 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 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 Fusion Middleware Web services.	See Chapter 5, "Managing Web Services on IBM WebSphere"	See Chapter 5, "Managing Web Services on IBM WebSphere"
Management of Oracle Platform Security Services (OPSS) features	See Chapter 6, "Managing Oracle Fusion Middleware Security on IBM WebSphere"	See Chapter 6, "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 a 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 hose, you will need to specify 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 on UNIX operating systems, use this command syntax:

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

To start on Windows operating systems, use this command syntax:

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

The following example demonstrates the command using 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 running the `OracleHelp.help()` command.

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

```
wsadmin>print OracleHelp.help()
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)
OracleMWConfig              Oracle Middleware Configuration Tool.
OracleMWConfigUtilities     Oracle Middleware Configuration Tool Utilities.
OracleODL                   Lists commands for FMW diagnostic logging.
OracleUMS                   Lists commands for User Messaging Service (UMS).
URLConnection               List Commands for managing ADF Based URL
                            Connections
WebServices                 Lists commands for Oracle WebServices Management.
audit                       Lists commands for Common Audit Framework
opss                        Oracle platform security services Commands.
soa                         Oracle platform SOA 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 quotes 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. For example, 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:

- Section 3.2.1, "Starting and Stopping Servers on IBM WebSphere"
- Section 3.2.2, "Configuring Metadata Services (MDS) on IBM WebSphere"
- Section 3.2.3, "Configuring Oracle Fusion Middleware Logging on IBM WebSphere"
- Section 3.2.4, "Setting Up the Diagnostic Framework"
- Section 3.2.5, "Creating a Data Source in an IBM WebSphere Cell"

3.2.1 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

3.2.1.1 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.

3.2.1.2 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 3.2.1.1, "Starting and Stopping IBM WebSphere Servers with Profile Scripts".

3.2.2 Configuring Metadata Services (MDS) on IBM WebSphere

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

Refer to the following sections for more information about the differences when running:

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

3.2.2.1 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, the `registerMetadataDBRepository` command has an additional parameter on IBM WebSphere (`authAlias`).

Also, the `targetServers` parameter allows you to specify a target WebSphere server or cluster for the repository. Note that if you later add additional servers or clusters to the cell, you must run the command again using the `targetServers` parameter to ensure that the repository is available on any new servers or clusters added after the initial registration of the repository.

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 `authAlias` parameter, 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, 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, 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 `authAlias` parameter is not provided.

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.

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, see Section 3.1.3.3, "Using the Oracle Fusion Middleware wsadmin Command-Line Online Help".

3.2.2.2 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.

3.2.3 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, wsadmin command line, or IBM WebSphere Administrative Console to modify the file.

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

  `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
```

3.2.4 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 can not 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.**

3.2.5 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

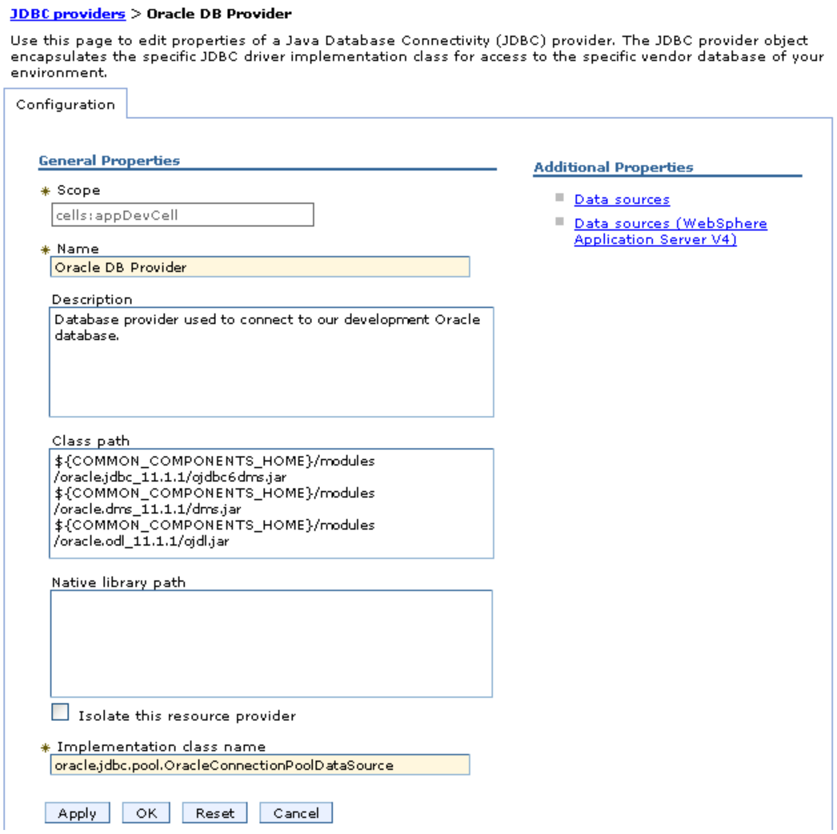
1. Click the name of the database provider in the list of JDBC providers.
2. In the General properties of the page, replace the value in the **Class path** field with the following:

   ```
   ${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



JDBC providers > **Oracle DB Provider**

Use this page to edit properties of a Java Database Connectivity (JDBC) provider. The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment.

Configuration

General Properties

* Scope

cells:appDevCell

* Name

Oracle DB Provider

Description

Database provider used to connect to our development Oracle database.

Class path

```
${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
```

Native library path

□ Isolate this resource provider

* Implementation class name

oracle.jdbc.pool.OracleConnectionPoolDataSource

Additional Properties

- Data sources
- Data sources (WebSphere Application Server V4)

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

1. Login 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, and WebCenter 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 applications, refer to the corresponding chapter in this guide and the 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.

Alternatively, you can use the MDS wsadmin commands to customize the MDS deployment plan before deploying the application archive from the IBM WebSphere Administrative Console.

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 Users and Groups in 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

4.1 Configuring Oracle SOA Suite Users and Groups in an External LDAP Server on IBM WebSphere

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:

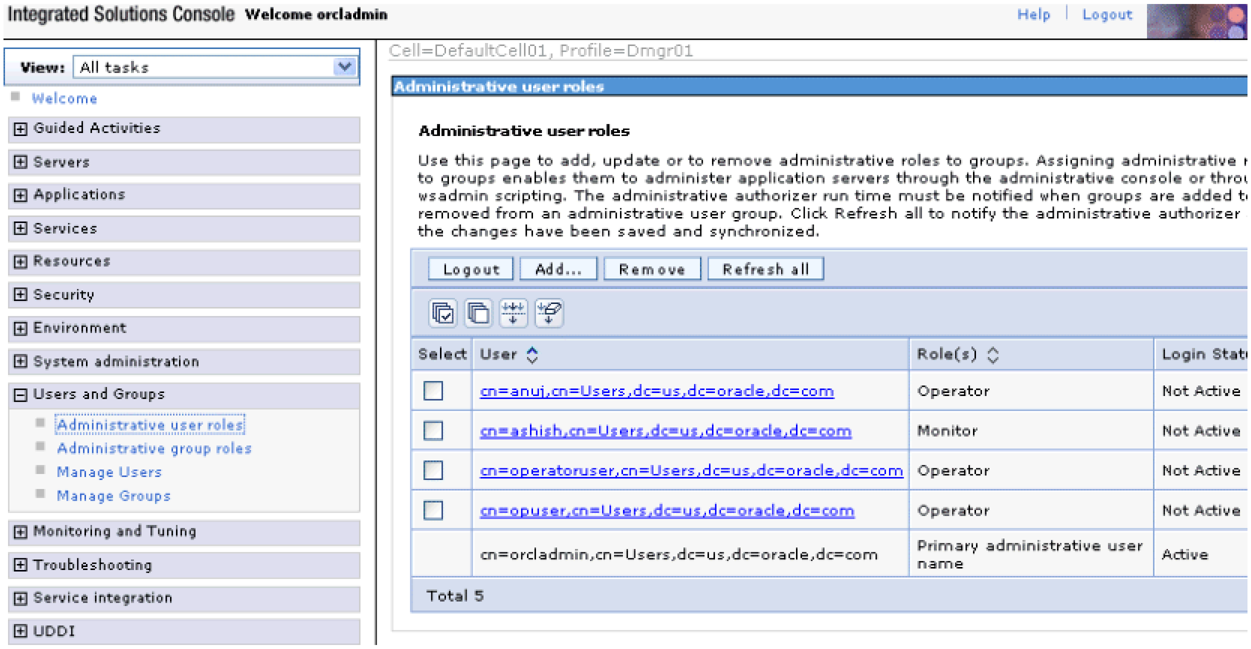
1. Use your LDAP management tool to create two groups (Operator user and Monitor user) and two users (Operator user and Monitor user).

   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 Administration 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 checkox) and the operator user **opuser** (fourth check box) available for selection. You perform similar mappings for group roles on a separate page.



Administrative user roles

Use this page to add, update or to remove administrative roles to groups. Assigning administrative r to groups enables them to administer application servers through the administrative console or throu wsadmin scripting. The administrative authorizer run time must be notified when groups are added t removed from an administrative user group. Click Refresh all to notify the administrative authorizer the changes have been saved and synchronized.

Select	User	Role(s)	Login Stat
□	cn=anuj,cn=Users,dc=us,dc=oracle,dc=com	Operator	Not Active
□	cn=ashish,cn=Users,dc=us,dc=oracle,dc=com	Monitor	Not Active
□	cn=operatoruser,cn=Users,dc=us,dc=oracle,dc=com	Operator	Not Active
□	cn=opuser,cn=Users,dc=us,dc=oracle,dc=com	Operator	Not Active
	cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com	Primary administrative user name	Active

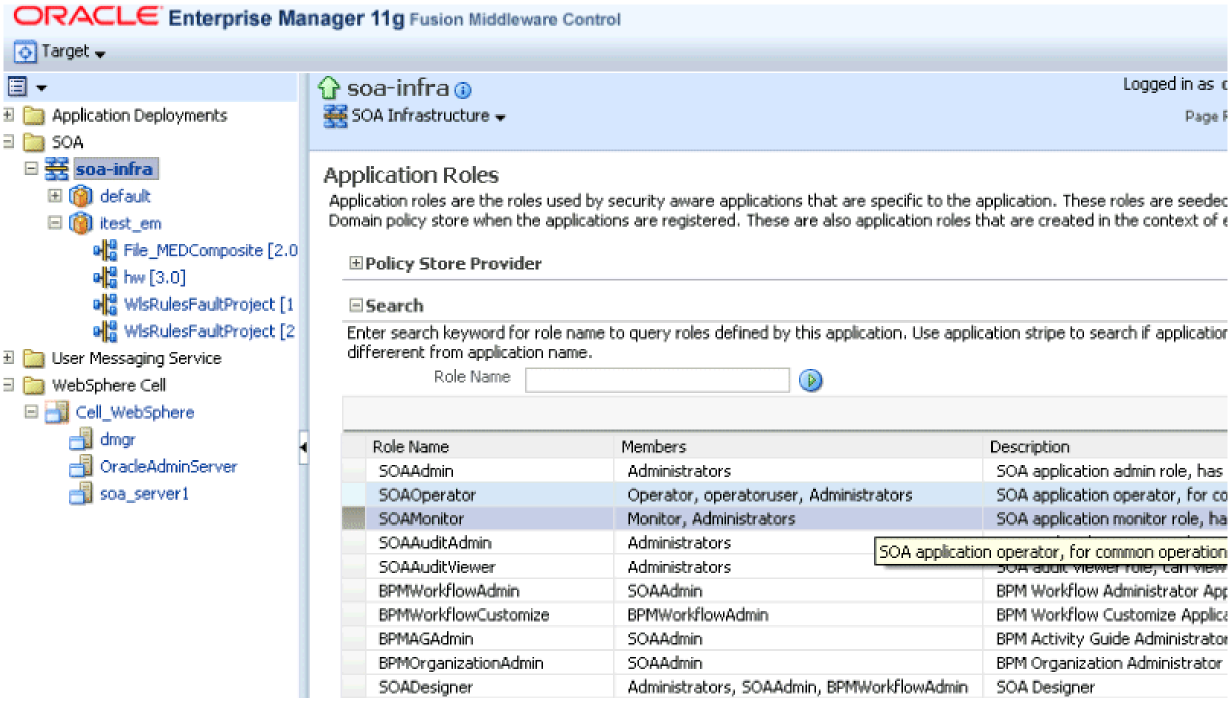
Total 5

3. Log in to Oracle Enterprise Manager Fusion Middleware Control Console 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.



Application Roles

Application roles are the roles used by security aware applications that are specific to the application. These roles are seedec Domain policy store when the applications are registered. These are also application roles that are created in the context of e

Enter search keyword for role name to query roles defined by this application. Use application stripe to search if application differerent from application name.

Role Name	Members	Description
SOAAdmin	Administrators	SOA application admin role, has
SOAOperator	Operator, operatoruser, Administrators	SOA application operator, for co
SOAMonitor	Monitor, Administrators	SOA application monitor role, ha
SOAAuditAdmin	Administrators	
SOAAuditViewer	Administrators	SOA audit viewer role, can view
BPMWorkflowAdmin	SOAAdmin	BPM Workflow Administrator App
BPMWorkflowCustomize	BPMWorkflowAdmin	BPM Workflow Customize Applica
BPMAGAdmin	SOAAdmin	BPM Activity Guide Administrator
BPMOrganizationAdmin	SOAAdmin	BPM Organization Administrator
SOADesigner	Administrators, SOAAdmin, BPMWorkflowAdmin	SOA Designer

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*.

4.2 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 *`ORACLE_HOME_for_SOA`*`/common/bin` directory:

```
cd ORACLE_HOME_for_SOA/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 11*g* Release 1 (11.1.1.4), apply fix pack 11).

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 hostname:

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

1. 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 username 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

1. 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

1. 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** > **JrfServer** > **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 Administration Console for which you are prompted. The location differs based on the type of IBM WebSphere Server you are using.

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**

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, *`WebSphere_Home`*`\bin\wsadmin.bat` for Windows or *`WebSphere_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, *`WebSphere_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, *`WebSphere_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:

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

  On UNIX:

  ```
  WebSphere_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=WeblogicDemoTrustStore
  com.ibm.ssl.trustStore=c:/YOUR_JDEVHOME/wlserver_
  10.3/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.

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 wasadmin
   -lang jython
   ```

   d. Enter the following command to create the credentials:

   ```
   Opss.createCred(map="SOA",key="Deployer",user="wasadmin",password="password
   ")
   ```

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 Administration 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 `wasadmin` user belongs).

h. Click **OK**.

i. Click the **Save** link.

j. Restart the server.

4.2.6.2 EJB Client

1. 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

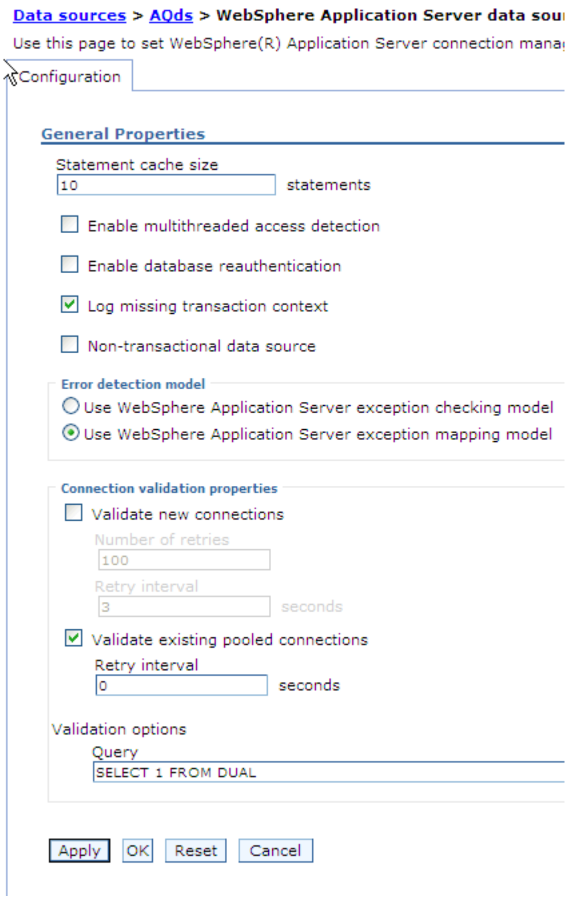
You must include the EJB stubs for the external EJB interface in the composite `SCA-INF/classes` or `SCA-INF/lib` directory.

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 WebSphere Application Server 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.



Data sources > AQds > WebSphere Application Server data sou

Use this page to set WebSphere(R) Application Server connection mana

Configuration

General Properties

Statement cache size: 10 statements

□ Enable multithreaded access detection

□ Enable database reauthentication

☑ Log missing transaction context

□ Non-transactional data source

Error detection model

○ Use WebSphere Application Server exception checking model

● Use WebSphere Application Server exception mapping model

Connection validation properties

□ Validate new connections

Number of retries: 100

Retry interval: 3 seconds

☑ Validate existing pooled connections

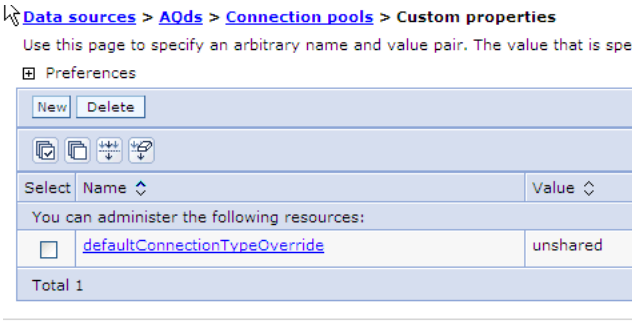
Retry interval: 0 seconds

Validation options

Query: SELECT 1 FROM DUAL

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.



Data sources > AQds > Connection pools > Custom properties

Use this page to specify an arbitrary name and value pair. The value that is spe

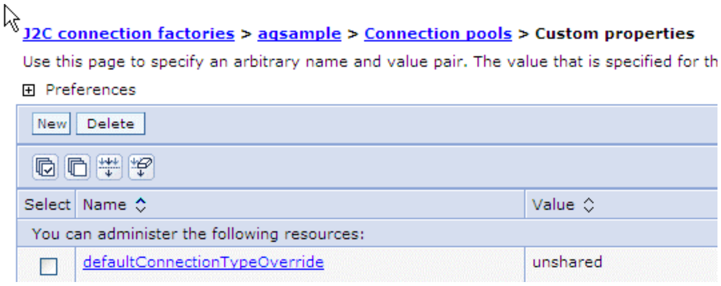
Select	Name	Value
You can administer the following resources:		
□	defaultConnectionTypeOverride	unshared
Total 1		

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.



J2C connection factories > aqsample > Connection pools > **Custom properties**

Use this page to specify an arbitrary name and value pair. The value that is specified for th

Select	Name	Value
You can administer the following resources:		
□	defaultConnectionTypeOverride	unshared

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.

4.2.9 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 Console on IBM WebSphere.

4.3.2 Oracle Enterprise Manager Fusion Middleware Control Console Shortcut Links

Oracle Enterprise Manager Fusion Middleware Control Console does not include shortcut links to the WebSphere Administration 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 Administration Console.

5

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:

- Section 5.1, "Configuring a Default Administrative User from the LDAP Directory"
- Section 5.2, "Configuring Oracle WSM on IBM WebSphere"
- Section 5.3, "Differences and Restrictions When Developing Web Services Applications on IBM WebSphere,"
- Section 5.4, "Differences and Restrictions When Managing Web Services Components on IBM WebSphere"
- Section 5.5, "Using the Web Services wsadmin Commands"

5.1 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 6, "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 6-1 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.

5.2 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

5.2.1 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 5-6.

> **Note:** **Oracle JRF for WebSphere** is automatically selected as a dependency when you select any of the above products.

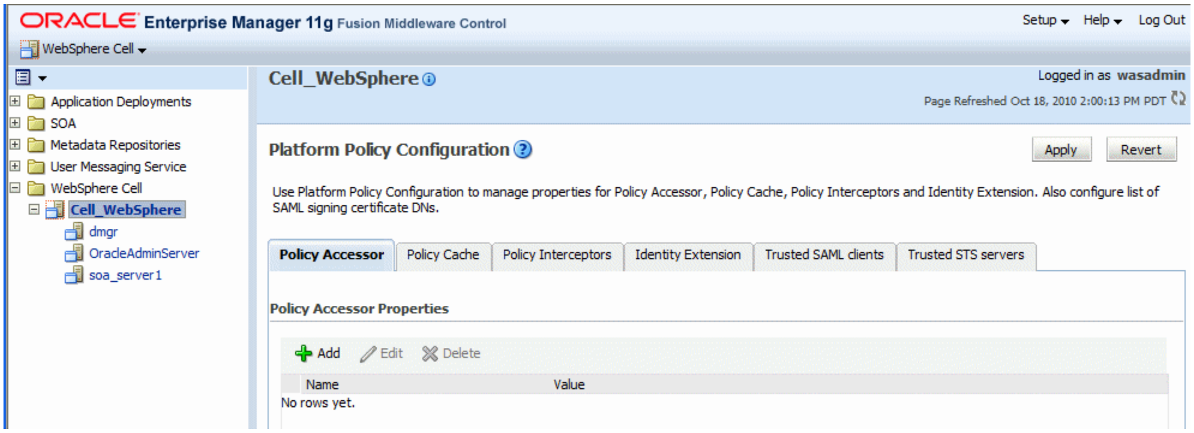
5.2.2 Connecting to the Oracle WSM Policy Manager

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 5–1.

Figure 5–1 Platform Policy Configuration



Use Platform Policy Configuration to manage properties for Policy Accessor, Policy Cache, Policy Interceptors and Identity Extension. Also configure list of SAML signing certificate DNs.

Name	Value
No rows yet.	

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 5-1.

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.

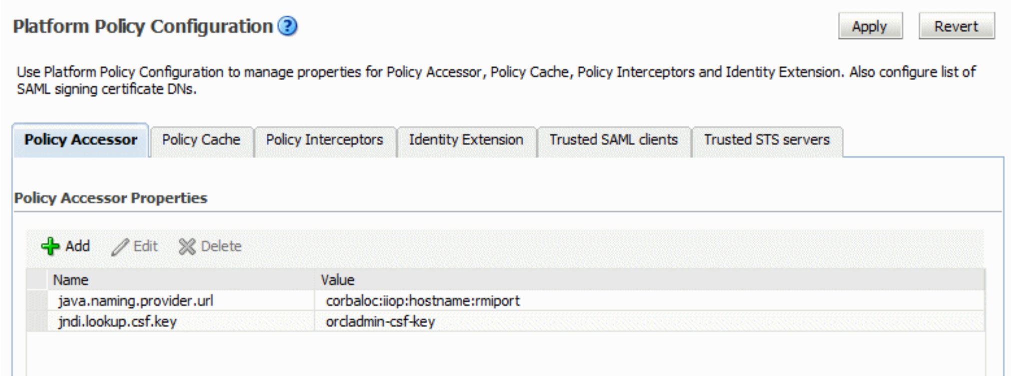
Because the Policy Manager is security enabled, the csf-key specifies the `java.naming.security.principal` and `java.naming.security.credentials` when using the JNDI URL to look up a Policy Manager.

For example, using the sample provided in "Configuring a Default Administrative User from the LDAP Directory" on page 5-1, the administrative user is `orcladmin` and the csf-key is `orcladmin-csf-key`.

c. Click **OK**.

Figure 5–2 shows the Policy Accessor tab with the `java.naming.provider.url` and `jndi.lookup.csf.key` property settings.

Figure 5–2 Policy Accessor Property Settings



Use Platform Policy Configuration to manage properties for Policy Accessor, Policy Cache, Policy Interceptors and Identity Extension. Also configure list of SAML signing certificate DNs.

Name	Value
java.naming.provider.url	corbaloc:iiop:hostname:rmiport
jndi.lookup.csf.key	orcladmin-csf-key

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.

5.3 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

5.3.1 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.

5.3.2 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 5-2. Once you have created or extended a cell with this template, the JMS queues are available for use.

5.3.3 JDeveloper

When using JDeveloper, the remote Oracle WSM policy store on a WebSphere server is not available.

5.4 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:

- Automatic Discovery of Oracle WSM Policy Manager
- Web Services Atomic Transactions
- No Support for Native Web Services
- Reliable Messaging
- Enterprise Manager Fusion Middleware Control

5.4.1 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 5-2.

5.4.2 Web Services Atomic Transactions

Web Services Atomic Transactions (WSAT) are not supported and will result in runtime errors.

5.4.3 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.

5.4.4 Reliable Messaging

WS-Reliable Messaging (WS-RM) is supported on IBM WebSphere with the following limitations:

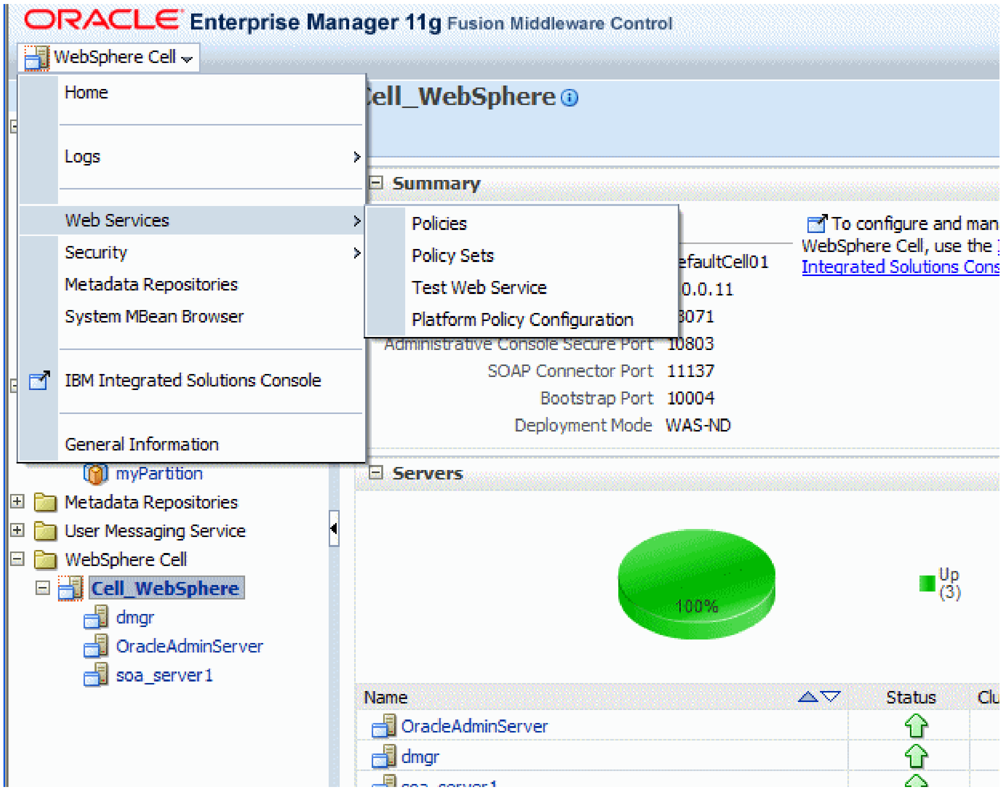
- WS-RM includes support for persistent database (DB) message store with Oracle databases only.
- WS-RM supports clustering only when Coherence is installed and available. This behavior is the same as WebLogic Server on all the platforms where Coherence is available.

5.4.5 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 5–3.

Figure 5–3 Web Services Menu



efaultCell01
0.0.11
3071
Administrative Console Secure Port 10803
SOAP Connector Port 11137
Bootstrap Port 10004
Deployment Mode WAS-ND

- 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.

5.5 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 5-9.
- 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 5-11.

> **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

5.5.1 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
                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
```

5.5.2 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 5-8.

Table 5–1 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 5–1 (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.

5.5.3 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 5-8.

Table 5–2 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.
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.
listPolicySets	Lists the policy sets in the repository.

Table 5–2 (Cont.) wsmManage Commands Supported on IBM WebSphere

Command	Description
modifyPolicySet	Specify an existing policy set to be modified in the current 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.

6

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:

- Section 6.1, "IBM WebSphere Identity Stores"
- Section 6.2, "Migrating Policies at Deployment"
- Section 6.3, "Migrating Credentials at Deployment"
- Section 6.4, "Reassociating Policies with reassociateSecurityStore"
- Section 6.5, "Deployment Mode"
- Section 6.6, "Configuring the JpsFilter and the JpsInterceptor"
- Section 6.7, "Using System Variables in Code Source URLs"
- Section 6.8, "Sample opss-application File"
- Section 6.9, "Executing Common Audit Framework wsadmin Commands"

6.1 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

6.1.1 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

6.1.2 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
```

6.2 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
```

6.2.1 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*.

6.2.2 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*.

6.2.3 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*.

6.3 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
```

6.3.1 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*.

6.4 Reassociating Policies with reassociateSecurityStore

For complete details about the scrip `reassociateSecurityStore` to reassociate the policy store, see *Oracle Fusion Middleware Application Security Guide*.

6.5 Deployment Mode

On IBM WebSphere, deployment is supported *only* in online mode; no offline deployment is supported.

6.6 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*.

6.7 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*.

6.8 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>
```

6.9 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*.

7 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:

- Section 7.1, "Introduction to OAM Identity Assertion on IBM WebSphere"
- Section 7.2, "Installing Components for the Oracle Access Manager IAP for IBM WebSphere"
- Section 7.3, "Introduction to the Oracle Access Manager 10g (10.1.4.3) Configuration Tool"
- Section 7.4, "Provisioning WebGate and Configuring OAM 10g (10.1.4.3) and the IAP for IBM WebSphere"
- Section 7.5, "Provisioning and Configuring OAM 11g for the IAP and IBM WebSphere"
- Section 7.6, "Installing the Required WebGate for the IHS Web Server"
- Section 7.7, "Preparing the IHS Web Server"
- Section 7.8, "Preparing the Login Form for WebGate"
- Section 7.9, "Configuring IBM WebSphere for OAM SSO and the IAP"
- Section 7.10, "Configuring SSO Logout for OAM IAP for IBM WebSphere"
- Section 7.11, "Known Issues"

7.1 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.

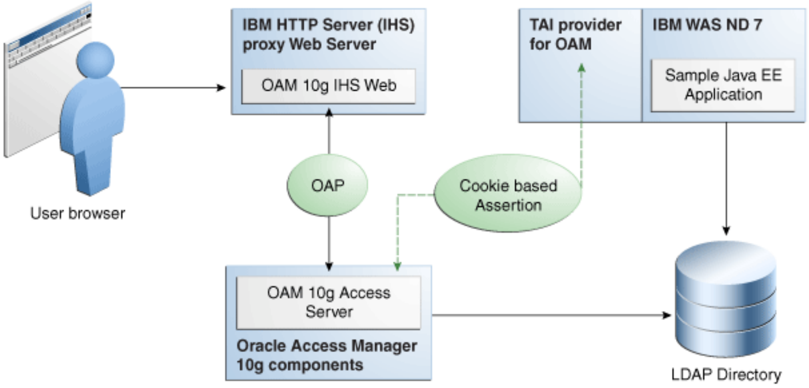
For more information, see the following topics:

- 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

7.1.1 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 7–1 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.

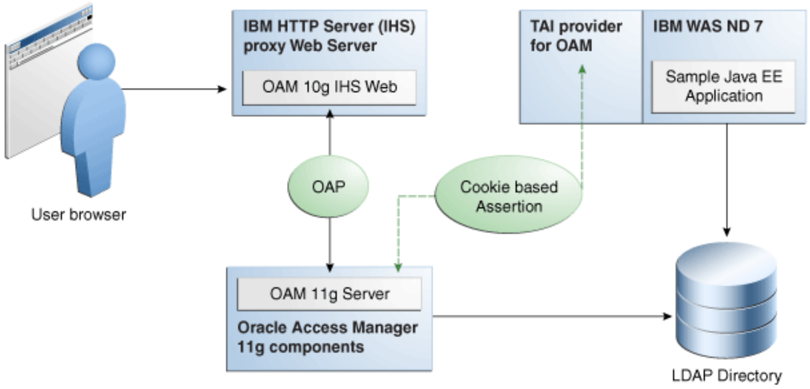
   IBM WebSphere checks for the existence of user in the user registry (configured LDAP instance) supplied by the OAM IAP. If the user is found, the assertion is successful. IBM WebSphere does not check for or request user's password in this scenario.

4. SSO Logout: See "Configuring SSO Logout for OAM IAP for IBM WebSphere" on page 7-20.

7.1.2 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 7–2 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 7-20.

7.2 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:

```
$MiddleWareHome/oracle_common/modules/oracle.oamprovider_11.1.1/
OAMTrustAssociationInterceptor.jar
```

Oracle Access Manager IAP for IBM WebSphere configuration file is located at:

```
$MiddleWareHome/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.

7.3 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 7–1 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 7–1 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 7–2 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 7–2 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 7–1. In this case, the return name OAM_REMOTE_USER is not configurable in oamtai.xml.

Table 7–1 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 7–2 Authorization Actions for "Header Based Assertion" in Protected URI Policy

Type	Name	Return Attribute
HeaderVar	OAM_REMOTE_USER	uid

7.4 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 7-6

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 7-11.

7.5 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

7.5.1 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 7–3 describes each of named text fields where you enter requested information on the Create OAM Agent page.

Table 7–3 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* for more information

7.5.2 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 7–3.
4. **Protected Resource List**: In this table, enter individual resource URLs to be protected by this OAM Agent, as shown in Table 7–3.
5. **Public Resource List**: In this table, enter individual resource URLs to be public (not protected), as shown in Table 7–3, 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".

7.6 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 7-13.

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 7-13.

7.7 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".

7.8 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".

7.9 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

7.9.1 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).

7.9.2 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".

7.9.3 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".

7.9.4 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:

```
$MiddleWareHome/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:

   ```
   $MiddleWareHome/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.

7.9.5 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

7.9.5.1 About Configuring the OAM TAI Configuration File

The oamtai.xml configuration file is available in the following path:

```
$MiddleWareHome/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 7–4 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 7–4 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 7–4 (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.

7.9.5.2 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 7–4 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 7–4).
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.

7.10 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

7.10.1 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 7–3 JavaScript to invoke "delOblixCookie" upon loading the page body. The LTPAToken is deleted by JavaScript; ObSSOCookie is deleted by WebGate.

```
<body onload="delOblixCookie();">
```

Example 7–3 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 7–3 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*.

7.10.2 Configuring Logout for ADF-Coded Applications

In ADF coded Fusion Middleware Applications such as Oracle WebCenter, 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

7.10.2.1 Configuring WebGate for Logout

This topic provides an example (Example 7–4) and procedure that you can use and customize to logout an application protected by OAM 10g with a 10g WebGate

> **Note:** Example 7–4 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 7–4: add and call the function `handleLogout()` for redirecting the logout request to the end URL specified in the logout URL

Example 7–4 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".

7.10.2.2 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.

7. Proceed to "Configuring oamAuthenProvider.jar in the IBM WebSphere classpath".

7.10.2.3 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 7-16.

The oamAuthnProvider.jar file is available from the following path:

```
$MiddleWareHome/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   $MiddleWareHome/oracle_common/modules/oracle.oamprovider_11.1.1/
   OAMTrustAssociationInterceptor.jar:$MiddleWareHome/oracle_common/modules/
   oracle.oamprovider_11.1.1/oamAuthnProvider.jar
   ```

5. Restart IBM WebSphere.
6. Proceed to "Verifying SSO Logout"

7.10.2.4 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.

7.11 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
- Deployments Region of the WebSphere Application Server Home Page

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

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.