Oracle® Web Services Manager

Deployment Guide

10*g* (10.1.3.4)

E12576-01

June 2008

Oracle Web Services Manager Deployment Guide, 10*g* (10.1.3.4)

E12576-01

Primary Author: Vrinda Kirloskar, Laureen Asato

Contents

C Oracle Web Services Manager WSMADMIN Commands

D Database Maintenance

E Authentication Sources

F Updating Host Names

Index

Preface

This guide describes enhancing the default Oracle Web Services Manager (Oracle WSM) environment. The topics covered include:

- Oracle WSM Topology
- Deployment Environments
- Oracle WSM Components and Application Servers Configuration
- Oracle WSM Database
- Transport Protocols
- Policy Enforcement Points Configuration
- Authentication Engine Configuration

> **Note:** Oracle Web Services Manager was previously known as Oblix COREsv and Confluent Core.

This Preface discusses the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This guide targets the needs of anyone who is responsible for deploying Web services management solutions. If you do not have Oracle WSM installed, see *Oracle Web Services Manager Installation Guide*.

This document assumes that you are familiar with networking concepts.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

How to Use This Guide

Use the *Oracle Web Services Manager Deployment Guide* for specific information on configuring and deploying the Oracle WSM components. This guide also covers securing Oracle WSM components over Secure Sockets Layer (SSL), integrating Oracle WSM with Oracle Access Manager, and configuring Oracle WSM in a clustered environment.

The *Oracle Application Server Enterprise Deployment Guide* describes how to distribute and deploy Services Oriented Architecture (SOA) components, including Oracle WSM, in an enterprise topology. The *Oracle Application Server Enterprise Deployment Guide* shows how Oracle WSM fits into an enterprise topology. However, the details of how to configure and deploy Oracle WSM components are found in Oracle Web Services Manager Deployment Guide.

Related Documents

For more information on Oracle WSM, see the following documents in the Oracle Web Services Manager 10*g* (10.1.3.4) documentation set:

- *Oracle Web Services Manager Installation Guide*—Provides instructions for installing and configuring the Oracle Web Services Manager components; it also details how to verify system operation and performance, and how to troubleshoot problems.
- *Oracle Web Services Manager Administrator's Guide*—Provides instructions for using the Web Services Manager Control to manage Web services as well as perform routine tasks to monitor Oracle Web Services Manager status and performance in a production environment.
- *Oracle Web Services Manager Extensibility Guide*—Provides information on extending Oracle Web Services Manager by creating and deploying new custom policy steps.

The following documents are referenced in this guide:

- *Oracle Application Server Installation Guide for Microsoft Windows*
- *Oracle Application Server Installation Guide for Linux x86*
- *Oracle Application Server Enterprise Deployment Guide*
- *Oracle Access Manager Installation Guide*
- *Oracle Access Manager Access System Administration Guide*
- *Oracle Access Manager Identity and Common Administration Guide*
- *Oracle Access Manager Developer Guide*
- *Oracle Containers for J2EE Security Guide*
- *Oracle HTTP Server Administrator's Guide*

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.
ORACLE_HOME	Directory in which the Oracle product is installed.

File Path Locations

When describing the location of files in this book, the UNIX convention of using a forward slash (/) to denote directories, is used. For example:

`ORACLE_HOME/owsm/config/gateway/gateway-config-installer.properties` file

If you are using Oracle Web Services Manager on a Windows operating system, replace the forward slashes with back slashes (\). For example:

`ORACLE_HOME\owsm\config\gateway\gateway-config-installer.properties` file

1

Planning An Oracle Web Services Manager Deployment

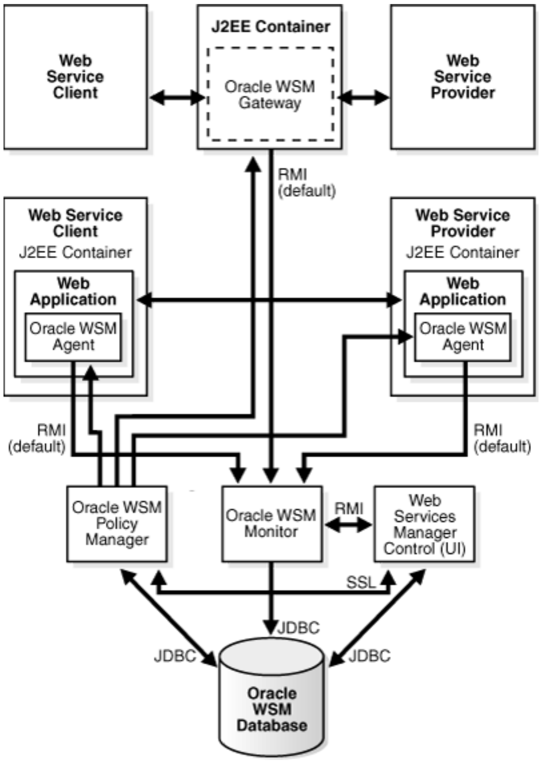
This chapter includes the following sections:

Oracle Web Services Manager Deployment

This section briefly describes a basic deployment of Oracle Web Services Manager (Oracle WSM). Figure 1–1 illustrates the components in an Oracle WSM deployment and the connections between these components:

- Policy Enforcement Points (Oracle WSM Agents and Oracle WSM Gateway)
- Oracle WSM Policy Manager
- Oracle WSM Monitor
- Oracle Enterprise Manager 10*g* Web Services Manager Control (Web Services Manager Control)
- Oracle WSM Database

Figure 1–1 Oracle WSM Deployment



Policy Enforcement Points

Oracle WSM enforces and manages security of Web services through runtime components called policy enforcement points (PEPs). There are two types of PEPs:

- **Oracle WSM Gateways** – The Oracle WSM Gateway is deployed in a J2EE container. It functions independently of the Web service it protects and acts as a proxy to the Web service clients.
- **Oracle WSM Agents** – There are two types of Oracle WSM Agents:
  - Client Agents
  - Server Agents

  Both types of agents run in the same space as the applications they protect, hosted by J2EE containers. Client agents intercept Web service requests from Web service clients, and enforce policy steps such as encryption or signature. Server agents intercept Web service requests before they reach the protected Web service and enforce access control security steps such as authentication and authorization.

Oracle WSM Administrative Components

The Oracle WSM administrative components are installed in a J2EE container and are used to manage an Oracle WSM environment:

- **Oracle WSM Policy Manager** – Oracle WSM Policy Manager manages Web service registration, policy configuration, and policy communication. The Oracle WSM Policy Manager stores policies in the Oracle WSM Database and uploads policies to the PEPs

- **Oracle WSM Monitor –** Oracle WSM Monitor manages the collection, aggregation, and persistence of data for monitoring Web services traffic. The Oracle WSM Monitor consists of two subcomponents: the Collector and the Aggregator. The Collector collects all the information coming from the PEPs during runtime, and the Aggregator applies aggregation rules, defining the information to be displayed in graphical charts.
- **Web Services Manager Control –** Web Services Manager Control is a Web-based application from which the Oracle WSM system is administered. It is the user interface to the Oracle WSM administrative components (Oracle WSM Policy Manager and Oracle WSM Monitor).

Oracle WSM Database

The Oracle WSM Database stores the following information:

- Security Policies
- Service-Level Agreements (SLA)
- Monitoring Data
- System Configuration

For high availability and scalability of your Oracle WSM environment, Oracle Real Application Clusters (RAC) is recommended.

Oracle WSM in Clustered Environments

Oracle WSM supports clustered environments. For more information on configuring an Oracle WSM cluster, see "Configuring Oracle WSM in a Clustered Environment" on page 2-4 and see *Oracle Application Server Enterprise Deployment Guide*.

Scaling An Oracle WSM Installation

Oracle WSM is designed to grow as Web services environments grow in size, diversity, and complexity. The administrative components can all be installed on one computer, or they can be distributed across multiple computers to optimize performance, security, monitoring, and distributed management.

Policy Enforcement Points (PEPs) can be redundant to distribute the load, improving throughput and reducing down time.

Multiple instances of the Oracle WSM Policy Manager can be installed to improve availability and throughput. Load balancers can be deployed to ensure the availability of the Oracle WSM Policy Manager in the case of a single server or host machine failure.

Use the following guidelines to scale the Oracle WSM Policy Manager:

- Place the Oracle WSM Policy Manager instances on separate application servers. The Oracle WSM Policy Manager is horizontally scalable.
- If a load balancer is placed between Web Services Manager Control and multiple instances of Oracle WSM Policy Manager, then the load balancer must be configured so that session affinity is supported for traffic to Oracle WSM Policy Manager. Session affinity is a method of configuring applications such that the client (in this example, Web Services Manager Control) is always connected to the same server (in this example, the application server on which the Oracle WSM

Policy Manager is installed). Session affinity ensures that the configuration changes for a session are sent to the same instance of Oracle WSM Policy Manager.

Multiple instances dedicated to a particular zone in the enterprise can also improve performance. For example, if the Web services environment is scattered across geographic zones, different instances can be used to monitor each zone separately.

2

Implementing Your Oracle Web Services Manager Deployment

This chapter provides an overview of an Oracle Web Services Manager (Oracle WSM) deployment. You must understand the concepts in this chapter before deploying the individual components. This chapter includes the following sections:

- Understanding the Oracle WSM Directory Structure
- Deploying Applications
- Localizing Oracle WSM
- Generating Log Files
- Configuring Oracle WSM Behind a Proxy Server
- Configuring Oracle WSM in a Clustered Environment

Understanding the Oracle WSM Directory Structure

If you install Oracle WSM using the default path, the installer creates the following directories:

- `ORACLE_HOME/owsm/config/` – This directory contains the source configuration files for your Oracle WSM installation. Any changes you make to the configuration files in this directory take effect when you redeploy the application.
- `ORACLE_HOME/j2ee/instance/applications` – The variable, *`instance`*, is the name of the OC4J instance into which Oracle WSM is installed. This directory contains the runtime files for your installation. Within this directory, there is a subdirectory for each Oracle WSM application. Any changes you make to the configuration files only affect the local runtime application.

Oracle recommends that you edit the source configuration files in the `ORACLE_HOME/owsm/config/` directory and deploy a component or application using these files. Your changes are saved and propagated to the runtime directory when you deploy the application. The reverse does not hold true. Any changes you make to configuration files in the runtime directory are *not* propagated back to the configuration files in the source directory.

The following section describes the recommended process for deploying applications and components.

Deploying Applications

After planning the topology of your Oracle WSM system, you must deploy the applications and components using the `wsmadmin` command-line tool. In most situations, you can deploy the applications and components using a single command, `wsmadmin deploy`. This command builds, deploys, and binds all Oracle WSM components or specified components to an OC4J application engine.

There may be situations where you need to separate the building and deploying of your applications or components. In this situation, use the `wsmadmin buildApps` command, followed by the `wsmadmin deployApps` command. See Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information about the `wsmadmin` commands.

Localizing Oracle WSM

Oracle WSM is localized in nine languages, in addition to the default language, English. These languages are automatically installed during the product installation. During the installation, the Oracle Universal Installer detects your locale. If your locale is one of the supported locales, the installer installs Oracle WSM in that locale. If the locale is not supported, then the Oracle Universal Installer installs Oracle WSM in English, the default locale.

The online help for the components (such as Oracle WSM Policy Manager, Oracle WSM Monitor, and the Web Services Manager Control) is also available in the nine supported locales.

Generating Log Files

Oracle WSM generates a log file for each running application. You can find these log files in the runtime directory under *`ORACLE_HOME`*`/j2ee/`*`instance`*`/log`. The *`instance`* variable is the name of the OC4J instance into which Oracle WSM is installed. If you have instances of the same application running concurrently, Oracle recommends that you create a separate log file for each instance.

Each application falls under a different category, and each category has different log levels.

The configuration parameters for the log files are loaded at runtime; therefore, it is better to adjust logging at the runtime level. By doing so, you can troubleshoot the specific application generating the error.

See *Oracle Web Services Manager Administrator's Guide* for more information about setting logging levels.

Configuring Oracle WSM Behind a Proxy Server

In a production environment, you may want to deploy Oracle WSM behind a proxy server. The proxy server provides indirect access to services on another machine, in this case to applications in your Oracle WSM system. The proxy server may cache previous requests to the Oracle WSM applications, allowing the applications to respond more quickly. For application requests that have not been cached or have expired, the proxy retrieves the information and forwards it to the requesting machine. Setting up Oracle WSM behind a proxy server has the following advantages:

- Performance is improved.

- A centralized request queue that adds another layer of security for Oracle WSM can be created.
- Access to specific URLs can be restricted.

Oracle WSM can be installed as a standalone installation or as part of Oracle Application Server 10g Release 3 (10.1.3.1.), referred to as the Oracle SOA Suite. See *Oracle Web Services Manager Installation Guide* for more information about the standalone installation, and see *Oracle Application Server Installation Guide for Microsoft Windows* and *Oracle Application Server Installation Guide for Linux x86* for more information about the Oracle SOA Suite installations. The following procedures describe how to configure proxy settings for each type of installation.

To configure a standalone Oracle WSM installation behind a proxy server

1. Open the following file:

   `ORACLE_HOME/owsm/bin/coresv.properties`

2. Edit the following parameters:

   ```
   proxy.host = proxy_server
   proxy.port = listen_port
   noproxy.hosts = localhost *domain.com
   ```

Table 2–1 Parameter Settings for Standalone Oracle WSM

Parameter	Description
proxy_server	Name of the proxy server, for example, www-proxy.us.oracle.com.
`listen_port`	The port number on the proxy server where you wish to connect. For example, 80.
localhost *domain.com	Hosts that connect directly without intervention from the proxy server. This value can be a list of host names separated by a vertical bar (|), or an asterisk (*), for example, `localhosts|*oracle.com`.

To configure Oracle WSM behind a proxy server when it is installed as part of Oracle SOA Suite

1. Open the following file:

   `ORACLE_HOME/opmn/conf/opmn.xml`

2. Find the `process-type id` whose value is the name of the instance in which Oracle Web Services Manager is installed. This may be `"home"`, or it could be another instance name. For example:

   ```
   ...
   <ias-component id="default_group">
      <process-type id="home" module-id="OC4J" status="enabled">
   ...
   ```

3. Find the `data id="java-options"` in the `category id="start-parameters"` section of the file.

   ```
   ...
   <category id="start-parameters">
      <data id="java-options" value="-server -XX:MaxPermSize=128M .../>
   </category>
   ...
   ```

4. Add the following parameters under java-options:

```
Dhttp.proxySet = true
Dhttp.proxyHost = proxy_server
Dhttp.proxyPort = listen_port
Dhttp.nonproxyHost = *localhost|*domain.com
```

Table 2–2 Parameter Settings for Oracle WSM Installed as Part of Oracle SOA Suite

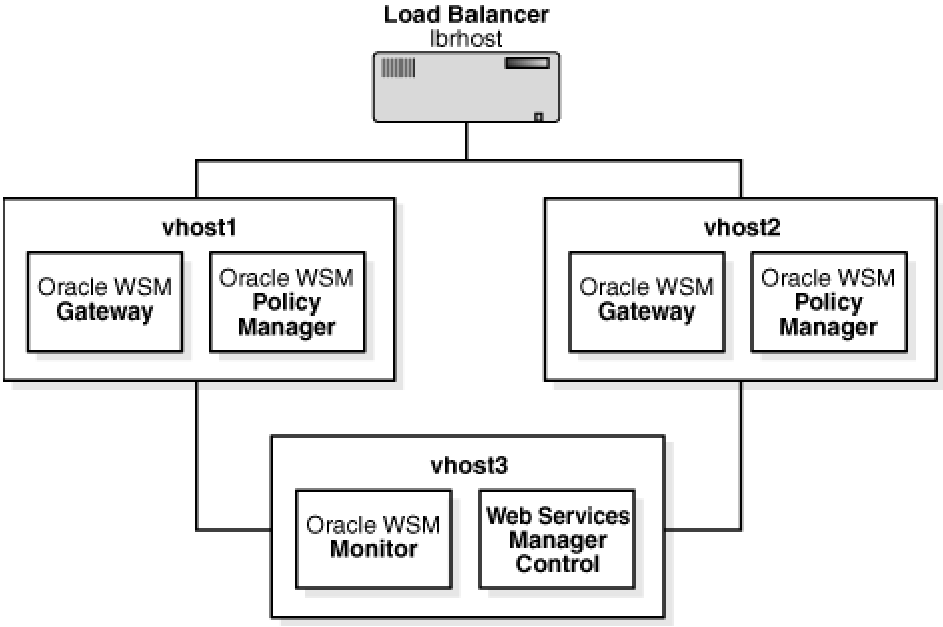
Parameter	Description
`true` or `false`	Set the value to true to enable the proxy server.
`proxy_server`	Name of the proxy server, for example, `proxy.mycompany.com`.
`listen_port`	The port number on the proxy server to which you wish to connect. For example, 80.
`*localhost|*domain.com`	Hosts that connect directly without intervention from the proxy server. This value can be a list of host names separated by a vertical bar (`|`), or an asterisk (`*`), for example, `localhosts|*mycompany.com`.

5. Restart the Oracle WSM Server for the configuration changes to take effect.

Configuring Oracle WSM in a Clustered Environment

The following is an example of Oracle WSM in a clustered environment. There are three instances of Oracle WSM, each installed on a different virtual host. The fourth virtual host acts as the load balancer. See Figure 2–1.

Figure 2–1 Example of Oracle Web Services Manager in a Clustered Environment



- Instance 1 and Instance 2 are the active-active instances with the following components:

  - Oracle WSM Policy Manager
  - Oracle WSM Gateway

  Instance 1 and Instance 2 are installed on `vhost1` and `vhost2`. Only one Oracle WSM Gateway can be installed in an Oracle WSM instance.

- Instance 3 includes the following components:
  - Oracle WSM Monitor
  - Web Services Manager Control

  Instance 3 is installed on `vhost3`.

- The fourth virtual host is a load balancer (`lbrhost`).

Oracle WSM components communicate with each other through a single endpoint URL. Therefore, if there are multiple instances of an Oracle WSM component—for example, multiple instances of the Oracle WSM Policy Manager—you must use a load balancer between the Oracle WSM components and the Web service client. In Figure 2–1, the load balancer provides the single endpoint URL to which client requests get made.

Configuration for all Oracle WSM components is static. Therefore, a computer cannot dynamically determine the host name or port of any computer, including the computer it resides on or the computer of any component with which it communicates. Machine addresses must be configured to reference virtual endpoints that always exist, and this requires the use of a load balancer between components.

The following sections describe the steps involved in setting up this example.

Installation Overview

Follow the instructions in the *Oracle Web Services Manager Installation Guide* for installing all three instances of Oracle WSM.

- Install all Oracle WSM components on instance 1, and specify `lbrhost` as the host.
- Install all Oracle WSM components on instance 2, and specify `lbrhost` as the host.
- Install all Oracle WSM components on instance 3, and specify `vhost3` as the host.

The Oracle WSM components are bundled together by default. Therefore, all components get installed. After you have completed the installation, you remove the components you do not need.

Setting up the Web Services Manager Control

In the following procedure, you set up the Web Services Manager Control (on `vhost3`) to communicate with the Oracle WSM Policy Manager (on `vhost1` and `vhost2`) by going through the load balancer (`lbrhost`).

1. On `vhost3`, edit the *`ORACLE_HOME`*`/owsm/config/ccore/ui-config-installer.properties` file.
2. Set the following properties:

Property Name	Value
`ui.pm.server.httpScheme`	`http` or `https`
`ui.pm.server.httpHost`	Oracle WSM Policy Manager host name. In this example, this is the load balancer, `lbrhost`.
`ui.pm.server.httpPort`	Oracle WSM Policy Manager HTTPS port. In this example, this is the lbrhost port.

3. Deploy the application using the `wsmadmin deploy control` command.

   See "deploy" in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information on this command.

> **Note:** Always perform this procedure first, before disabling the Oracle WSM components, to avoid errors in the log file.

Disabling Oracle WSM Components on vhost3

When you install Oracle WSM, by default, all components are installed. You need to remove the Oracle WSM Policy Manager and Oracle Web Services Manager Gateway components from `vhost3`. Before you proceed with the following steps, verify that you have set up the Web Services Manager Control first.

1. On `vhost3`, disable the Oracle WSM Policy Manager component using the `wsmadmin undeploy policymanager` command.
2. On `vhost3`, disable the Oracle Web Services Manager Gateway component using the `wsmadmin undeploy gateway` command.

See "undeploy" in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information on this command.

Setting up the Oracle WSM Gateway

To set up the Oracle WSM Gateway, you need to register the gateway from the Web Services Manager Control. By registering the gateway, you are creating a logical component with policies. Web services are secured by creating a virtual endpoint through which all requests must be routed. Complete the following tasks to set up the gateway:

- Connect the gateway to the Oracle WSM Monitor on vhost3.
- Configure the gateway with its component ID.
- Redeploy the gateway.

The procedures that follow describe how to complete these tasks.

To set up the Oracle WSM Gateway

> **Note:** Perform steps 1 through 3 only once for each logical gateway.

1. Log in to Web Services Manager Control on `vhost3`.

2. Register the Oracle WSM Gateway.

    a. In the navigation pane, click **Policy Management,** then click **Manage Policies.**

    b. Click **Add New Component.**

    c. From the Component Type list, select Gateway.

    d. In the Component URL box, enter the URL for the load balancer (`lbrhost`).

    e. Click **Register.**

    Remember to make a note of the component ID. This ID will be used in step 4.

    See "Prerequisites and General Information about Registering Oracle WSM Gateways" on page 5-1 for more information about registering gateways.

3. Connect the Oracle WSM Gateway with the Oracle WSM Monitor on `vhost3`.

    a. In the navigation pane of Web Services Manager Control, click **Policy Management,** then click **Manage Policies.**

    b. Locate the gateway in the List of Components, and click the **Edit** icon for that gateway.

    c. Set the following properties for the Oracle WSM Monitor, then click **Save.**

Property Name	Value
`cfluent.monitor.rmi.host`	Host on which the Oracle WSM Monitor resides. In this example, this is vhost3.
`cfluent.monitor.rmi.port`	Oracle WSM Monitor RMI port.

> **Note:** Perform steps 4 and 5 once for each physical gateway.

4. Configure the Oracle WSM Gateway with its component ID.

    a. Open the following file on `vhost1`:

    ORACLE_HOME`/owsm/config/gateway/`
    `gateway-config-installer.properties`

    b. In the following line, replace the component ID with the one you noted in step 2.

    ```
    gateway.component.id=C0003001
    ```

5. Redeploy the Oracle WSM Gateway on `vhost1` using the `wsmadmin deploy gateway` command.

    See "deploy" in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information on this command.

Disabling Oracle WSM Components on vhost1 and vhost2

Each Oracle Web Services Manager installation has all components running (that is, Oracle WSM Policy Manager, Oracle WSM Monitor, Oracle WSM Gateway, and Web Services Manager Control). However, you can only have one instance of Oracle WSM Monitor in your entire Oracle Web Services environment. Therefore, you must disable the Oracle WSM Monitor on `vhost1` and `vhost2`.

1. On vhost1, disable Oracle WSM Monitor by executing the `wsmadmin undeploy monitor` command.
2. On vhost1, disable Web Services Manager Control by executing the `wsmadmin undeploy control` command.
3. Repeat steps 1 and 2 for vhost2.

See "undeploy" in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information on this command.

Testing the Oracle WSM System

Requests for the Web service come through the load balancer. The load balancer then directs the request to vhost1 or vhost2. Check to see if you can access the WSDL by going through the load balancer (lbrhost).

1. Register the Web service to the Oracle WSM Gateway.
   a. In the navigation pane, click **Policy Management,** then click **Register Services.**
   b. Click **Add New Service.**
   c. Register the WSDL for the service you want to protect.
   d. Click **Finish,** then click **OK.**

   See *Oracle Web Services Manager Administrator's Guide* for more information on registering a Web service to the gateway.
2. Get the URL for the WSDL.
   a. From the navigation pane of the Web Services Manager Control, click **Policy Management,** then click **Register Services.**
   b. Click the **Services** link for the gateway.
   c. Click **View Details** icon for the Web service.
   d. Copy the URL in the **Service WSDL URL** field.
3. Test whether you can access the Web service using the protected URL.
   a. In the navigation pane of Web Services Manager Control, click **Tools,** then click **Test Page.**
   b. Paste the URL you copied into the **Enter wsdl url** box, and click **Submit Query.** The page refreshes and displays other parameters you can set.
   c. Click **Invoke.** You should get a Test Result page.

   See *Oracle Web Services Manager Administrator's Guide* for more information on using the Test Page.

What Happens After This?

Once you have read and understood the concepts in this chapter, you can configure and deploy the individual Oracle WSM components. For information about configuring the Oracle WSM components, refer to Chapter 3, "Configuring the Oracle Web Services Manager Components" on page 3-1.

3

Configuring the Oracle Web Services Manager Components

This chapter provides general information for modifying the configuration files for the Oracle Web Services Manager (Oracle WSM) components: Oracle Enterprise Manager 10*g* Web Services Manager Control (Web Services Manager Control), Oracle WSM Monitor, Oracle WSM Agents, Oracle WSM Gateways, and Oracle WSM Policy Manager.

See Chapter 5, "Deploying Oracle WSM Gateways", for specific information about configuring Oracle Web Services Manager Gateways.

See Chapter 6, "Installing Oracle WSM Agents", for specific information about configuring Oracle Web Services Manager Agents.

This chapter includes the following sections:

- Deploying Components
- Oracle WSM Component Configuration Files
- Configuring Oracle WSM Database Connections

Deploying Components

Use the `wsmadmin` command with the appropriate operation, that is, `configApps`, `deployApps` or `deploy`, to configure and deploy the customized components. See "Deploying Applications" on page 2-2 for more information about the `wsmadmin` commands to use. See Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for information on the individual commands.

Component Context Root Strings

The portion of the URL address that immediately follows the server name and port specification is called the context root.

```
http://server_name:port/context_root/....
```

The `context_root` is mapped to the J2EE application. The HTTP Server uses this mapping to locate the application. Table 3–1 lists the valid context root strings for Oracle WSM components.

Table 3–1 Oracle WSM Context Root Strings

Component Name	Context Root String
Oracle WSM Policy Manager	`policymanager`
Web Services Manager Control	`ccore`

Table 3–1 (Cont.) Oracle WSM Context Root Strings

Component Name	Context Root String
Oracle WSM Monitor	`coreman`
Oracle WSM Gateway	`gateway`

An example URL for the Web Services Manager Control is: `http://jdoe.us.oracle.com:8888/ccore`.

If you use `wsmadmin` commands to deploy your applications, the correct context root strings are used, by default. However, if you use Oracle Enterprise Manager 10*g* Application Server Control Console to deploy your components, there are no defaults. You must specify the context root strings yourself, and you must use the context root strings in Table 3–1 to ensure that Oracle WSM works properly.

Changing Component Properties

The following is a list of properties that can and cannot be changed:

Properties that can be changed:

- Oracle WSM Database instance associated with the Oracle WSM component
- Oracle WSM Database connections for multiple components in the `*ORACLE_HOME*/owsm/config/bin/coresv.properties` file

Properties that cannot be changed:

- Host operating system
- Host application server

Oracle WSM Component Configuration Files

The following sections list the configuration files and identify the properties in each file used to configure each Oracle WSM component.

Related properties in each configuration file are grouped together. For example, in the `monitorui-config-installer.properties` file, the properties used to configure the database drivers and URLs are grouped together. In the sections that follow, these groupings are loosely referred to as *property groups*.

Web Services Manager Control Configuration Files

To configure the Web Services Manager Control, you need to modify certain properties files. The default location for the configuration files for the Web Services Manager Control is `*ORACLE_HOME*/owsm/config/ccore`. The three configuration files that contain properties relevant to the Web Services Manager Control are as follows:

- `monitorui-config-installer.properties`
- `policyui-config-installer.properties`
- `ui-config-installer.properties`

The monitorui-config-installer.properties File

The default location for the `monitorui-config-installer.properties` file for the Web Services Manager Control is `*ORACLE_HOME*/owsm/config/ccore`. This section describes properties in the `monitorui-config-installer.properties` file. The properties in this file are grouped together as follows:

- Database Drivers and URLs
- UI Authentication
- Support Contact
- Oracle WSM Monitor Repository
- Web Services Manager Control and Oracle WSM Monitor Pairing
- Corda
- Notification Engine

This section also contains information about how to enable Java Single Sign-On (JSSO) for the Oracle WSM Monitor. See "Enabling JSSO for the Oracle WSM Monitor" on page 3-6, for more information.

Database Drivers and URLs

The database drivers and URLs properties are used to configure the database URLs and drivers. Table 3–2 lists the properties and provides an example value for each property.

Table 3–2 monitorui-config-installer.properties Database Drivers and URLs Settings

Property	Example
`sqlserverUrl`	`jdbc:microsoft:sqlserver://${dbhost}:${dbPort};SelectMethod=cursor;databaseName=${dbInstance}`
`sqlserverDriver`	`com.microsoft.jdbc.sqlserver.SQLServerDriver`
`oracleThinUrl`	`jdbc:oracle:thin:@//${dbhost}:${dbPort}/${service}`
`oracleThinDriver`	`oracle.jdbc.driver.OracleDriver`
`oracleThickURL`	`jdbc:oracle:oci:@${dbAlias}`
`oracleThickDriver`	`oracle.jdbc.driver.OracleDriver`
`oliteThinUrl`	`jdbc:polite4@${dbHost}:${dbPort}:${dbInstance}`
`oliteThinDriver`	`oracle.lite.poljdbc.POLJDBCDriver`

UI Authentication

The UI authentication properties are used to configure the authentication provider used by Web Services Manager Control. You can use a database or an LDAP server to authenticate users. Table 3–3 lists the properties and provides an example value for each property in the Database Authentication Settings property group.

Table 3–3 monitorui-config-installer.properties Database-based Authentication Settings

Property	Example
`ui.authentication.provider`	`${dbAuthenticationProvider}`
`ui.authentication.provider.properties`	`\`
`dbConnectionUrl`	`jdbc:polite$@localhost:3120:orawsm`

Table 3–3 (Cont.) monitorui-config-installer.properties Database-based Authentication

Property	Example
`dbDriver`	`oracle.lite.poljdbc.POLJDBCDriver|\`
`dbUser`	`system|\`
`dbPassword`	`manager|\`
`maxConnections`	`5|\`
`idleTime`	`300|\`
`maxConnectionTime`	`120|\`

Table 3–4 lists the properties and provides an example value for the LDAP authentication settings properties.

Table 3–4 monitorui-config-installer.properties LDAP-based Authentication Settings

Property	Example
`ui.authentication.provider`	`${ldapAuthenticationProvider}`
`ui.authentication.provider .properties`	`\`
`ldapHost`	`192.168.1.214|\`
`ldapPort`	`389|\`
`ldapDN`	`ou=People,dc=corp,dc=confluentsw,dc=com|\`
`superUserRole`	`SystemAdmin|\`
`roleAttribute`	`groupmembership`

Support Contact

You can specify an e-mail address for your support contact. Use the following property to specify the fully qualified e-mail address of your support contact:

```
ui.supportContact.email= support_contact_email_address
```

Oracle WSM Monitor Repository

The Oracle WSM Monitor Repository properties are used to configure the Web Services Manager Control Repository. Table 3–5 lists the properties and provides examples values.

Table 3–5 monitorui-config-installer.properties Oracle WSM Monitor Repository Settings

Property	Example
`ui.repository.url`	`jdbc:polite4@localhost:1531:orawsm`
`ui.repository.driver`	`oracle.lite.poljdbc.POLJDBCDriver`
`ui.repository.userid`	`system`
`ui.repository.password`	This is the property that contains the encoded password for the repository.
`ui.repository.maxConnections`	`5`
`ui.repository.maxConnectionTime`	`120`

Table 3–5 (Cont.) monitorui-config-installer.properties Oracle WSM Monitor Repository Settings

Property	Example
`ui.repository.idleConnectionTime`	`1000`
`ui.componentRepository.url`	`jdbc:polite4@localhost:1531:orawsm`
`ui.componentRepository.driver`	`oracle.lite.poljdbc.POLJDBCDriver`
`ui.componentRepository.userid`	`system`
`ui.componentRepository.password`	This property contains the encoded password for the component repository.
`ui.componentRepository.maxConnections`	`5`
`ui.componentRepository.maxConnectionTime`	`120`
`ui.componentRepository.idleConnectionTime`	`1000`

Oracle WSM Monitor and Web Services Manager Control Pairing

The Oracle WSM Monitor communicates with and exchanges information with the Web Services Manager Control. The two properties you can use to set up the pairing between these two components are `ui.om.server.rmiHost` and `ui.om.server.rmiPort`.

Corda

Table 3–6 lists properties and description of the Corda settings.

Table 3–6 monitorui-config-installer.properties Corda Settings

Property	Description
`ui.corda.mode`	The valid values for this property are `library` or `server`.
`ui.corda.libraryMode.contextRoot`	If you have set `ui.corda.mode` to `library`, provide a value for this setting.
`ui.corda.serverMode.redirectServlet`	If you have set `ui.corda.mode` to `server`, provide a value for this setting.

Notification Engine

The Notification Engine properties are used to collect information used for business analysis. Table 3–7 lists the properties and description of each property.

Table 3–7 monitorui-config-installer.properties Notification Engine Settings

Property	Description
`ui.notification.smtp.from`	The value for this property defaults to the e-mail address in `ui.supportContact.email`.
`ui.notification.snmp.toolkit`	Leave this property blank if you want to use the default toolkit.
`ui.notification.snmp.mgrHost`	The value for this property is the DNS name for the management software.

Table 3–7 (Cont.) monitorui-config-installer.properties Notification Engine Settings

Property	Description
`ui.notification.snmp.mgrPort`	This property contains the port number where traps will be sent. The default value is 162.
`ui.notification.snmp.authProtocol`	The value for this property is the authentication protocol for the management software.
`ui.notification.snmp.user`	The value for this property is the user name to authenticate on traps.
`ui.notification.snmp.Password`	The value for this property is the password to authenticate on traps.

Enabling JSSO for the Oracle WSM Monitor

The *Oracle Web Services Manager Installation Guide* describes how to enable JSSO for the Oracle WSM Monitor, by editing the `install.properties` file. Follow this procedure to enable JSSO for the Oracle WSM Monitor.

To enable JSSO for the Oracle WSM Monitor

1. Open the *`ORACLE_HOME`*`/owsm/bin/install.properties` file, and verify that the following parameter is set:

   ```
   install.sso.support=true
   ```

2. Go to *`ORACLE_HOME`*`/owsm/bin` and run the `wsmadmin deploy` command:

   ```
   wsmadmin deploy control
   ```

The policyui-config-installer.properties File

The default location for the `policyui-config-installer.properties` file for the Web Services Manager Control is *`ORACLE_HOME`*`/owsm/config/ccore`. This section describes properties in the `policyui-config-installer.properties` file. The properties in this file are grouped together as follows:

- Database Drivers and URLs
- UI Authentication
- Support Contact
- Oracle WSM Monitor
- Oracle WSM Monitor and Oracle WSM Policy Manager Pairing

Database Drivers and URLs

Use these settings to configure the database URLs and drivers. Table 3–8 lists properties and examples of values in the Database Drivers property group of the `policyui-config-installer.properties` file:

Table 3–8 policyui-config-installer.properties Database Drivers and URLs Settings

Property	Example
`sqlserverUrl`	`jdbc:microsoft:sqlserver://${dbhost}:${dbPort};SelectMethod=cursor;databaseName=${dbInstance}`

Table 3–8 (Cont.) policyui-config-installer.properties Database Drivers and URLs

Property	Example
`sqlserverDriver`	`com.microsoft.jdbc.sqlserver.SQLServerDriver`
`oracleThinUrl`	`jdbc:oracle:thin:@//${dbhost}:${dbPort}/${service}`
`oracleThinDriver`	`oracle.jdbc.driver.OracleDriver`
`oracleThickURL`	`jdbc:oracle:oci:@${dbAlias}`
`oracleThickDriver`	`oracle.jdbc.driver.OracleDriver`
`oliteThinUrl`	`jdbc:polite4@${dbHost}:${dbPort}:${dbInstance}`
`oliteThinDriver`	`oracle.lite.poljdbc.POLJDBCDriver`

UI Authentication

Use these settings to configure the authentication provider used by the Web Services Manager Control. Currently you can select database-based authentication or LDAP-based authentication. Table 3–9 lists properties and examples of values in database-based authentication property group of the `policyui-config-installer.properties` file.

Table 3–9 policyui-config-installer.properties Database-based Authentication Settings

Property	Example
`ui.authentication.provider`	`${dbAuthenticationProvider}`
`ui.authentication.provider.properties`	`\`
`dbConnectionUrl`	`jdbc:polite$@localhost:3120:orawsm`
`dbDriver`	`oracle.lite.poljdbc.POLJDBCDriver|\`
`dbUser`	`system|\`
`dbPassword`	`manager|\`
`maxConnections`	`5|\`
`idleTime`	`300|\`
`maxConnectionTime`	`120|\`

Table 3–10 lists properties and examples of values in LDAP-based authentication property group of the `policyui-config-installer.properties` file.

Table 3–10 policyui-config-installer.properties LDAP-based Authentication Settings

Property	Example
`ui.authentication.provider`	`${ldapAuthenticationProvider}`
`ui.authentication.provider.properties`	`\`
`ldapHost`	`192.168.1.214|\`
`ldapPort`	`389|\`
`ldapDN`	`ou=People,dc=corp,dc=confluentsw,dc=com|\`

Table 3–10 (Cont.) policyui-config-installer.properties LDAP-based Authentication

Property	Example
`superUserRole`	`SystemAdmin|\`
`roleAttribute`	`groupmembership`

Support Contact

You can specify an e-mail address for your support contact. Use the following property to specify the fully qualified e-mail address of your support contact:

```
ui.supportContact.email=support_contact_email_address
```

Web Services Manager Control Repository

Use these property settings to configure the Web Services Manager Control Repository. Table 3–11 lists properties and examples of values in Web Services Manager Control Repository property group of the `policyui-config-installer.properties` file:

Table 3–11 policyui-config-installer.properties Web Services Manager Control Repository Settings

Property	Example
`ui.componentRepository.url`	jdbc:polite4@localhost:1531:orawsm
`ui.componentRepository.driver`	oracle.lite.poljdbc.POLJDBCDriver
`ui.componentRepository.userid`	`system`
`ui.componentRepository.password`	The value for this property contains the encoded password for the component repository.
`ui.componentRepository.maxConnections`	5
`ui.componentRepository.maxConnectionTime`	`120`
`ui.componentRegistry.idleConnectionTime`	`1000`

Oracle WSM Monitor and Oracle Web Services Manager Policy Manager Pairing

The `policyui-config-installer.properties` file lists properties and values in the Oracle WSM Monitor and Web Services Manager Control Pairing property group.

The ui-config-installer.properties File

The default location for the `ui-config-installer.properties` file for the Web Services Manager Control is *`ORACLE_HOME`*`/owsm/config/ccore`. This section describes properties in the `ui-config-installer.properties` file. The properties in this file are grouped together as follows:

- Database Drivers and URLs
- UI Authentication
- Support Contact
- Administrative Repository
- Oracle WSM Monitor
- Corda
- Notification Engine

Database Drivers and URLs

Use these settings to configure the database URLs and drivers. Table 3–12 lists properties and examples of values in the Database Drivers and URLs property group of the `ui-config-installer.properties` file:

Table 3–12 ui-config-installer.properties Database Drivers and URLs Settings

Property	Example
`sqlserverUrl`	`jdbc:microsoft:sqlserver://${dbhost}:${dbPort};SelectMethod=cursor;databaseName=${dbInstance}`
`sqlserverDriver`	`com.microsoft.jdbc.sqlserver.SQLServerDriver`
`oracleThinUrl`	`jdbc:oracle:thin:@//${dbhost}:${dbPort}/${service}`
`oracleThinDriver`	`oracle.jdbc.driver.OracleDriver`
`oracleThickURL`	`jdbc:oracle:oci:@${dbAlias}`
`oracleThickDriver`	`oracle.jdbc.driver.OracleDriver`
`oliteThinURL`	`jdbc:polite4@${dbHost}:${dbPort}:${dbInstance}`
`oliteThinURL`	`oracle.lite.poljdbc.POLJDBCDriver`

UI Authentication

Use these settings to configure the authentication provider used by the Web Services Manager Control. Currently you can select database-based authentication or LDAP-based authentication. Table 3–13 lists parameters and examples of values in the database-based authentication provider property group of the `ui-config-installer.properties`.

Table 3–13 ui-config-installer.properties Database-based Authentication Settings

Property	Example
`ui.authentication.provider`	`${dbAuthenticationProvider}`
`ui.authentication.provider.properties`	`\`
`dbConnectionUrl`	`jdbc:polite$@localhost:3120:orawsm`
`dbDriver`	`oracle.lite.poljdbc.POLJDBCDriver|\`
`dbUser`	`system|\`
`dbPassword`	`manager|\`
`maxConnections`	`5|\`
`idleTime`	Ignore this parameter. It is obsolete.
`maxConnectionTime`	Ignore this parameter. It is obsolete.

Table 3–14 lists parameters and examples of values in the LDAP-based authentication provider property group of the `ui-config-installer.properties`.

Table 3–14 ui-config-installer.properties LDAP-based Authentication Settings

Property	Example
`ui.authentication.provider`	`com.cluent.accessprovider.ldap.BasicLdapAuthProvider`

Table 3–14 (Cont.) ui-config-installer.properties LDAP-based Authentication Settings

Property	Example
`ui.authentication.provider.properties`	`\`
`ldapHost`	`acme.company.com|\`
`ldapPort`	`389|\`
`ldapSSLEnabled`	`false|\`
`ldapDN`	`o=acme,ou=company,c=us|\`
`roleAttribute`	`uniqueMember`

Support Contact

You can specify an e-mail address for your support contact. Use the following property, and specify the fully qualified e-mail address of your support contact:

```
ui.supportContact.email=support_contact_email_address
```

Administrative Repository

Use these property settings to configure the Web Services Manager Control Repository. Table 3–15 lists parameters and examples of values in the Administrative Repository property group of the `ui-config-installer.properties`.

Table 3–15 ui-config-installer.properties Administrative Repository Settings

Property	Example
`ui.repository.url`	`jdbc:polite4@localhost:1531:orawsm`
`ui.repository.driver`	`oracle.lite.poljdbc.POLJDBCDriver`
`ui.repository.userid`	`system`
`ui.repository.password`	The value for this property contains the encoded password for the repository.
`ui.repository.maxConnections`	`5`
`ui.repository.maxConnectionTime`	`120`
`ui.repository.idleConnectionTime`	`1000`
`ui.componentRepository.url`	`jdbc:polite4@localhost:1531:orawsm`
`ui.componentRepository.driver`	`oracle.lite.poljdbc.POLJDBCDriver`
`ui.componentRepository.userid`	`system`
`ui.componentRepository.password`	The value for this property contains the encoded password for the component repository.
`ui.componentRepository.maxConnections`	`5`
`ui.componentRepository.maxConnectionTime`	`120`
`ui.componentRepository.idleConnectionTime`	`1000`

Oracle WSM Monitor

The ui-config-installer.properties file lists parameters and values in the Oracle WSM Monitor property group. The two properties you can use to set up Oracle WSM Monitor are ui.om.server.rmiHost and ui.om.server.rmiPort.

Corda

These values can be set in the Corda property group of the ui-config-installer.properties file.

Notification Engine

The Notification Engine properties are used to collect information used for business analysis. Use these property settings to configure the notification engine. Table 3–16 lists properties and examples of values in the Notification property group of the ui-config-installer.properties file.

Table 3–16 ui-config-installer.properties Notification Engine Settings

Property	Description
ui.notification.smtp.from	The value for this property defaults to address in ui.supportContact.email.
ui.notification.snmp.toolkit	Leave the value for this property blank if you want to use the default toolkit.
ui.notification.snmp.mgrHost	The value for this property contains the DNS name for the management software
ui.notification.snmp.mgrPort	This property contains the port number where traps will be sent. The default value is 162.
ui.notification.snmp.authProtocol	This is the authentication protocol for the management software.
ui.notification.snmp.user	The value for this property is the user name to authenticate on traps.
ui.notification.snmp.Password	The value for this property is the password used to authenticate traps.

Oracle WSM Monitor Configuration Files

To configure the Oracle WSM Monitor, you need to modify certain properties files. The default location for the configuration files for the Oracle WSM Monitor is *ORACLE_HOME*/owsm/config/coreman. The two configuration files that contain properties relevant to the Oracle WSM Monitor are as follows:

- collector-config-installer.properties
- monitor-config-installer.properties

The collector-config-installer.properties File

The default location for the collector-config-installer.properties file for the Oracle WSM Monitor is *ORACLE_HOME*/owsm/config/coreman. Table 3–17 lists properties and examples of values in the collector-config-installer.properties file.

Table 3–17 collector-config-installer.properties Database Drivers and URLs Settings

Property	Example
sqlserverUrl	jdbc:microsoft:sqlserver://${dbhost}:${dbPort};SelectMethod=cursor;databaseName=${dbInstance}
sqlserverDriver	com.microsoft.jdbc.sqlserver.SQLServerDriver
oracleThinUrl	jdbc:oracle:thin:@//${dbhost}:${dbPort}/${service}
oracleThinDriver	oracle.jdbc.driver.OracleDriver
oracleThickURL	jdbc:oracle:oci:@${dbAlias}
oracleThickDriver	oracle.jdbc.driver.OracleDriver
oliteThinUrl	jdbc:polite4@${dbHost}:${dbPort}:${dbInstance}
oliteThinDriver	oracle.lite.poljdbc.POLJDBCDriver

The monitor-config-installer.properties File

The default location for the `monitor-config-installer.properties file` for the Oracle WSM Monitor is *ORACLE_HOME*`/owsm/config/coreman`. The properties in this file are grouped together as follows:

- Database Driver
- Oracle WSM Monitor Repository
- Oracle WSM Monitor and Web Services Manager Control Pairing
- Notification Engine

Database Drivers and URLs

Use these settings to configure the database URLs and drivers. Table 3–18 lists properties and examples of values in the Database Driver property group of the `monitor-config-installer.properties` file:

Table 3–18 monitor-config-installer.properties Database Drivers and URLs Settings

Property	Example
sqlserverUrl	jdbc:microsoft:sqlserver://${dbhost}:${dbPort};SelectMethod=cursor;databaseName=${dbInstance}
sqlserverDriver	com.microsoft.jdbc.sqlserver.SQLServerDriver
oracleThinUrl	jdbc:oracle:thin:@//${dbhost}:${dbPort}/${service}
oracleThinDriver	oracle.jdbc.driver.OracleDriver
oracleThickURL	jdbc:oracle:oci:@${dbAlias}
oracleThickDriver	oracle.jdbc.driver.OracleDriver
oliteThinUrl	jdbc:polite4@${dbHost}:${dbPort}:${dbInstance}
oliteThinDriver	oracle.lite.poljdbc.POLJDBCDriver

Oracle WSM Monitor Repository

Use these property settings to configure the Web Services Manager Control Repository. Table 3–19 lists properties and examples of values in the Oracle WSM Monitor Repository property group of the `monitor-config-installer.properties` file.

Table 3–19 monitor-config-installer.properties Oracle WSM Monitor Repository Settings

Property	Example
`monitor.repository.url`	`jdbc:polite4@localhost:1531:oraws m`
`monitor.repository.driver`	`oracle.lite.poljdbc.POLJDBCDriver`
`monitor.repository.userid`	`system`
`monitor.repository.password`	The value for this property contains the encoded password for the repository.

Oracle WSM Monitor and Web Services Manager Control Pairing

The Web Services Manager Control communicates with the Oracle WSM Monitor. The `monitor-config-installer.properties` file lists properties and values in the Oracle WSM Monitor and Web Services Manager Control Pairing property group.

Table 3–20 lists properties and examples of values in Oracle WSM Monitor and Web Services Manager Control Pairing group of the `monitor-config-installer.properties` file:

Table 3–20 monitor-config-installer.properties Oracle WSM Monitor and Web Services Manager Control Pairing Settings

Property	Example
`monitor.rmiServer.enabled`	`true`
`monitor.rmiServer.host`	`localhost`
`monitor.rmiServer.port`	`3118`

Notification Engine

The Notification Engine properties are used to collect information used for business analysis. Use these property settings to configure the notification engine. Table 3–21 lists properties and examples of values in the Notification property group of the `monitor-config-installer.properties` file.

Table 3–21 monitor-config-installer.properties Notification Engine Settings

Property	Description
`ui.notification.smtp.from`	The value defaults to the e-mail address in the `ui.supportContact.email` property.
`ui.notification.snmp.toolkit`	Leave this blank if you want to use the default toolkit.
`ui.notification.snmp.mgrHost`	This property contains the DNS name for the management software.
`ui.notification.snmp.mgrPort`	This property is the Port number where traps will be sent. The default value is 162.

Table 3–21 (Cont.) monitor-config-installer.properties Notification Engine Settings

Property	Description
`ui.notification.snmp.user`	This property contains the authentication protocol for the management software.
`ui.notification.snmp.password`	This property contains the user name to authenticate on traps.
`ui.external.help.location`	This property contains the password to authenticate on traps.

Oracle WSM Policy Manager Configuration Files

To configure the Oracle WSM Policy Manager, you modify properties in a single configuration file, *ORACLE_HOME*/owsm/config/policymanager/policymanager-config-installer.properties. This section contains a brief description of this property file.

The policymanager-config-installer.properties file

The default location for the `policymanager-config-installer.properties` file for the Oracle WSM Policy Manager is *ORACLE_HOME*/owsm/config/policymanager. The properties in this file are grouped together as follows:

- Oracle WSM Policy Manager Repository
- Oracle WSM Policy Manager Component Repository

Policy Repository

Use these property settings to configure the Oracle WSM Policy Manager Repository. Table 3–22 lists properties and examples of values in the Oracle WSM Policy Manager Repository property group of the `policymanager-config-installer.properties` file.

Table 3–22 policymanager-config-installer.properties Policy Repository Settings

Property	Example
`ui.componentRepository.url`	`jdbc:polite4@localhost:1531:orawsm`
`ui.componentRepository.driver`	`oracle.lite.poljdbc.POLJDBCDriver`
`ui.componentRepository.userid`	system
`ui.componentRepository.password`	This property contains the encoded password for the component repository.
`ui.componentRepository.maxConnections`	5
`ui.componentRepository.maxConnectionTime`	120
`ui.componentRepository.idleConnectionTime`	1000

Oracle WSM Policy Manager Component Repository

Use these property settings to configure the Oracle WSM Policy Manager Repository. Table 3–23 lists properties and examples of values in the Oracle WSM Policy Manager

Component Repository property group of the `policymanager-config-installer.properties` file.

Table 3–23 policymanager-config-installer.properties Component Repository Settings

Property	Example
`policymanager.componentRepository.url`	`jdbc:polite4@localhost:1531:orawsm`
`policymanager.componentRepository.driver`	`oracle.lite.poljdbc.POLJDBCDriver`
`policymanager.componentRepository.userid`	`system`
`policymanager.componentRepository.password`	This property contains the password for the component repository.

Oracle WSM Gateway Configuration Files

The default location for the Oracle Web Services Manager Gateway configuration file is *`ORACLE_HOME/`*`owsm/config/gateway/gateway-config-installer.properties`. The properties in this file are grouped together as follows:

- Component ID
- Administrative Repository
- Transport Protocol

Server Agent Configuration Files

The default location for the Oracle WSM server agent configuration file is *`ORACLE_HOME`*`/owsm/config/serveragent/serveragent-config-installer.properties`. The properties in this file are grouped together as follows:

- Component ID
- Container Type
- Policy Manager
- Policy Settings

Client Agent Configuration Files

The default location for the Oracle WSM client agent configuration file is *`ORACLE_HOME`*`/owsm/config/clientagent/clientagent-config-installer.properties`. The properties in this file are grouped together as follows:

- Component ID
- Container Type
- Policy Manager
- Policy Settings

Configuring Oracle WSM Database Connections

You can modify the *ORACLE_HOME*/owsm/config/bin/coresv.properties files to specify database connections for multiple components. Table 3–24 lists properties, property descriptions, and sometimes provides examples of values to configure database connections:

Table 3–24 Database Connection Parameters in the coresv.properties File

Property	Values and Descriptions
db.type	Indicates the type of database: oracle, sqlserver, or olite.
db.driver.type	Indicates the type of database driver: thin or thick.
db.host	The machine name where the database is installed, or the IP address.
db.port	This is the jdbc port.
db.name	The name of the Oracle WSM Database (default is CCORE, when it is an Oracle Database).
db.userid	The user name to log in to the Oracle WSM Database.
db.password	The password to log in to the Oracle WSM Database.
dataload.generateSql.dbType	The same value as db.type.
dataload.messagelog.db.url	The URL for the JDBC driver associated with the database. For example, jdbc:polite4@localhost:1531:owsm
dataload.messagelog.db.driver	If using OLite, the only valid value is thin. For Oracle Database 10*g*, specify thick or thin.
dataload.messagelog.db.userid	The administrator account to log in to the database. The default is system.
dataload.messagelog.db.password	The password associated with the account specified by dataload.messagelog.db.userid
dataload.messagelog.db.sid	The name of the database instance where the Oracle WSM Repository is stored. (The default is CCORE).

Configuring E-Mail Notification

Oracle WSM Monitor captures and stores service details, status, and performance metrics from each service request or response handled by Oracle WSM gateways or agents. You can configure Oracle WSM so that specific conditions trigger an alarm. A processing rule, which is a response to the alarm, is then executed. One possible response is to send an e-mail to the appropriate individuals. In order for the e-mail to be sent, you must configure Oracle WSM with the e-mail server and account information. See Chapter 6, "Monitoring Oracle Web Services Manager, in *Oracle Web Services Manager Administrator's Guide* for more information about sending an e-mail notification when an alarm is triggered.

To configure Oracle WSM for e-mail notification

1. Edit the properties in the *ORACLE_HOME*/owsm/config/ccore/ui-config-installer.properties file (Table 3–25).

Table 3–25 ui-config-installer.properties File

Property Name	Description
ui.notification.smtp.host	Name of the E-mail server that sends the e-mail notifications.
ui.notification.smtp.port	SMTP port on which the E-mail server sends the e-mail notifications.
ui.notification.smtp.user	User name of the account used to send e-mail notifications.
ui.notification.smtp.password	Password of the account used to send e-mail notifications. This password must be obfuscated. Use the wsmadmin encodePasswords command to generate an obfuscated password.
ui.notification.smtp.from	E-mail account from which the e-mail notifications are sent.

2. Edit the properties in the *ORACLE_HOME*/owsm/config/coreman/monitor-config-installer.properties file (Table 3–26).

Table 3–26 monitor-config-installer.properties File

Property Name	Description
monitor.notification.smtp.host	Name of the E-mail server that sends the e-mail notifications.
monitor.notification.smtp.port	SMTP port on which the E-mail server sends the e-mail notifications.
monitor.notification.smtp.user	User name of the account used to send e-mail notifications.
monitor.notification.smtp.password	Password of the account used to send e-mail notifications. This password must be obfuscated. Use the wsmadmin encodePasswords command to generate an obfuscated password.
monitor.notification.smtp.from	E-mail account from which the e-mail notifications are sent.

3. Edit the properties in the *ORACLE_HOME*/owsm/config/ccore/monitorui-installer.properties file (Table 3–27).

Table 3–27 monitor-ui-config-installer.properties File

Property Name	Description
ui.notification.smtp.host	Name of the E-mail server that sends the e-mail notifications.
ui.notification.smtp.port	SMTP port on which the E-mail server sends the e-mail notifications.
ui.notification.smtp.user	User name of the account used to send e-mail notifications.
ui.notification.smtp.password	Password of the account used to send e-mail notifications. This password must be obfuscated. Use the wsmadmin encodePasswords command to generate an obfuscated password.
ui.notification.smtp.from	E-mail account from which the e-mail notifications are sent.

4. Redeploy the Web Services Manager Control and the Oracle WSM Monitor components using the wsmadmin deploy command.

Note: See Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information on the `wsmadmin deploy` and `wsmadmin encodePasswords` commands.

4 Securing Oracle Web Services Manager Components over SSL

Oracle Web Services Manager (Oracle WSM) components send requests to and receive responses from other Oracle WSM components. To secure these communications, you can configure Oracle WSM components to use Secure Sockets Layer (SSL), the most widely used transport-level data-communication protocol providing:

- Authentication – Communication is established between two trusted parties.
- Confidentiality – The data exchange is encrypted.
- Message integrity – The data is checked for possible corruption.

This chapter includes the following sections:

- Introduction to Secure Communications in the Oracle WSM Environment
- Securing Oracle WSM Components
- Frequently Asked Questions

Introduction to Secure Communications in the Oracle WSM Environment

Oracle WSM Policy Manager, Oracle WSM Monitor, and Web Services Manager Control are J2EE applications hosted by the OC4J server. Therefore, Oracle WSM supports all of the security methods supported on the Java platform. See *Oracle HTTP Server Administrator's Guide* for more information on how to integrate the Oracle HTTP Server and the OC4J server.

Security Configuration Options

How you configure your transport security depends on the Oracle WSM package you installed and how the Oracle WSM components are managed. The following sections describe how to configure transport security based on the type of Oracle WSM installation and deployment.

Oracle SOA Suite Installation

If you installed Oracle WSM as part of the Oracle SOA Suite, then your OC4J server is managed by Oracle Process Manager and Notification Server (OPMN). See *Oracle Containers for J2EE Security Guide* for information about configuring OPMN to enable HTTPS and use SSL.

When you install the Oracle SOA Suite, there are two installation options. Depending on which option you selected, Oracle HTTP Server may or may not enabled, by default:

- If you installed Oracle WSM using the Basic installation option, then Oracle HTTP Server is not enabled, by default.
- If you installed Oracle WSM using the Advanced installation option, then Oracle HTTP Server is *not* enabled, by default. See Chapter 10, "Enabling SSL for the Oracle HTTP Server", in the *Oracle HTTP Server Administrator's Guide,* to enable Oracle HTTP Server.

See *Oracle Application Server Installation Guide for Microsoft Windows* or *Oracle Application Server Installation Guide for Linux x86* for more information about the different Oracle SOA Suite installation options.

Standalone Oracle WSM Installation

If you installed Oracle WSM as part of the standalone package, how the OC4J Server is managed depends on which of the following installation options you selected:

- In a Basic standalone installation, the OC4J server is not managed by OPMN.
- In an Advanced standalone installation, Oracle WSM is added to an existing installation of the Oracle SOA Suite, and the OC4J Server is managed by OPMN.

See *Oracle Web Services Manager Installation Guide* for more information about the different standalone installation options.

Summary of Oracle WSM Deployment Types

Table 4–1 summarizes the different Oracle WSM installations and the deployment type. The installation package and the installation option determines the type of Oracle WSM deployment. In "Configuring the Truststore in SSL Connections" the deployment type determines how you configure your secure communications.

Table 4–1 Deployment Types for Oracle WSM Installations

Installation Package	Installation Option	Oracle WSM Deployment Type
SOA Suite	Basic	OPMN-managed
	Advanced	OPMN-managed
Standalone	Basic	Standalone OC4J
	Advanced	OPMN-managed

Configuring the Truststore in SSL Connections

For any SSL connection between two components, the component that initiates the SSL connection acts as a client, and the client end of the connection must be configured to use a truststore. A truststore is a keystore file that includes the trusted certificate authorities that a client will implicitly accept during an SSL handshake. For example, if you want to secure a connection from Oracle WSM Gateway to Oracle WSM Policy Manager, then the gateway must be configured to use a truststore. This section describes how to configure the truststore file for clients in the following Oracle WSM deployments:

- OPMN-managed deployments
- Standalone OC4J deployments

See "Security Configuration Options" to determine which type of deployment you have installed.

OPMN-Managed Deployments

Perform this procedure if your Oracle WSM installation is managed by OPMN.

1. Modify the *`ORACLE_HOME`*`/opmn/conf/opmn.xml` file.
2. Add the JSSE truststore properties as system properties for the node with the XPath `/opmn/process-manager/ias-component[id="default-group"]/process-type[id="home"]/module-data/category[id=start-parameters"]/data[id="java-options"]` as follows:

   ```
   -Djavax.net.ssl.trustStore=c:/mykeystore.jks

   -Djavax.net.ssl.trustStorePassword=changeit

   -Djavax.net.ssl.trustStoreType=JKS
   ```

   Table 4–2 describes the JSSE truststore parameters.

Table 4–2 Description of JSSE Truststore Properties

Parameter	Description
`-Djavax.net.ssl.trustStore`	Path to the location of the truststore.
`-Djavax.net.ssl.trustStorePassword`	Password used to access the truststore.
`-Djavax.net.ssl.trustStoreType`	Type of file used for the truststore. Valid values are JKS or PKCS12.

Standalone OC4J Server Deployments

Perform this procedure if you have installed Oracle WSM as part of the standalone package, with the Basic installation option.

1. Modify the *`ORACLE_HOME`*`/owsm/scripts/oc4j.xml` file.
2. Add the JSSE truststore properties as system properties to the ant target "`oc4j.start`" as follows:

   ```
   <target name="oc4j.start" description="-->Start OC4J server">

   <echo message="Starting stand-alone OC4J server in
   ${oc4J.j2ee.home}"\>

   <java.jar="${oc4j.j2ee.home}\oc4.jar

   fork="true"

   failonerror='true"

   dir="${oc4j.j2ee.home}">

   <jvmarg value="-XX:MaxPermSize=128M"\>

   <jvmarg value="-Xms512M\>

   <jvmarg value="-Xmx512M"\>

   <sysproperty key="http.nonproxyHosts"
   value=&quot;${noproxy.hosts}&quot;/>

   <sysproperty key="http.proxyHost" value=${proxy.host}"/>

   <sysproperty key="http.proxyPort" value=${proxy.port}"/>

   <sysproperty key="javax.net.ssl.trustStoreType" value="JKS"/>
   ```

```
<sysproperty key="javax.net.ssl.trustStore"
value="c:\mykeystore.jks"/>

<sysproperty key="javax.net.ssl.trustStorePassword"
value="changeit"/>

</java>

</target>
```

Table 4–3 describes the JSSE Truststore parameters.

Table 4–3 Description of JSSE Truststore Parameters

Parameters	Description
`javax.net.ssl.trustStoreType`	Type of file used for the truststore. Valid values are JKS or PKCS12.
`javax.net.ssl.trustStore`	Path to the location of the truststore.
`javax.net.ssl.trustStorePassword`	Password used to access the truststore.

Securing OC4J Using SSL

Oracle WSM Gateway, Oracle WSM Policy Manager, and Web Services Manager Control are all J2EE applications on OC4J, and connections to them can be secured using SSL. The procedure below is an overview of how you secure these applications. For more information, see the chapter "SSL Communication with OC4J, in *Oracle Containers for J2EE Security Guide*.

To secure OC4J using SSL

1. Create the following file: `ORACLE_HOME/j2ee/instance/config/secure-web-site.xml`

   The variable *instance* is the name of your Oracle instance.

2. Add the `protocol="http"` and `secure="true"` properties to the `<web-site>` element.

   If connections to the Oracle WSM component pass through an Oracle HTTP Server, then the protocol must be set to Apache JServ Protocol (AJP). Add the `protocol="ajp13"` and `secure="true"` properties in the `<web-site>` element for these components.

3. Add the `<ssl-config>` element beneath the `<web-site>` element and define the keystore location and password, using the `keystore` and `keystore-password` attributes.

   For example:

```
<web-site display-name="OC4J Web Site" protocol="http" port="636" secure="true"
>
  <default-web-app application="default" name="defaultWebApp" root="/j2ee" />
  <access-log path="../log/default-web-access.log" />
  <ssl-config keystore="../keystore" keystore-password="welcome" />
</web-site>
```

4. For each Oracle WSM component that you want to enable with SSL, enter the `<web app>` entry. For example:

   - `<web app application="policymanager"...>`
   - `<web app application="gateway" ...>`

- `<web app application="ccore"...>`

> **Note:** You may copy the entry for the desired components from the `OC4J_HOME/j2ee/instance/config/default-web-site.xml` file

5. For each `<web app>` entry that you added to the `secure-web-site.xml` file, remove the `<web app>` entry for that same component from the `default-web-site.xml` file.
6. Restart Oracle Application Server.

Generally, each Oracle WSM component is specified in either the `default-web-site.xml` file or the `secure-web-site.xml` file in the `<web app>` element. That is, the component is either configured to accept requests on a secure protocol or an open protocol. If you wish to accept requests for a component on both protocols, then you must add the `shared="true"` property to the `<web-app>` element in both the `default-web-site.xml` and `secure-web-site.xml` files.

Configuring Two-Way SSL

For information on configuring Oracle WSM components for two-way SSL, also known as client authentication SSL, refer to *Oracle Containers for J2EE Security Guide* and *HTTP Server Administrator's Guide*.

Securing Oracle WSM Components

Figure 4–1 is a basic deployment of Oracle WSM and shows which components communicate with each other. The arrows start where the connection is initiated and end where the connection terminates. For example, one connection is initiated by the Web Service Client to the Oracle WSM Gateway. Table 4–4 describes each of the connections, the type of information being transmitted, and the supported transport protocols. For those connections that can be secured using SSL, Table 4–4 provides a link to the section in this chapter that describes how to configure the secure connection.

Figure 4–1 Oracle WSM Deployment Showing Component Connections

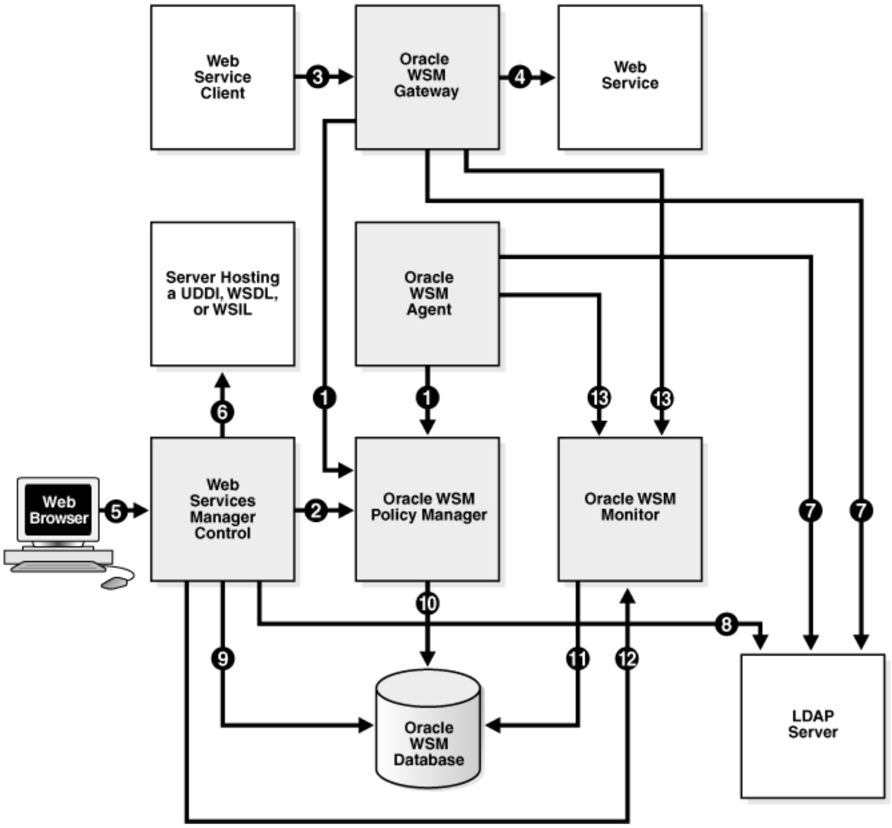


Table 4–4 Description of Connections Between Oracle WSM Components

Figure Number	Description of the Connection Between Components	Supported Secure Transport Protocols	For Information on How to Configure Secure Connections
1	The Policy Enforcement Points initiate an SSL connection over HTTP with Oracle WSM Policy Manager. The PEPs retrieve policy information and use it to enforce security on Web services.	SSL over HTTP	"Securing PEPs to Oracle WSM Policy Manager" on page 4-7
2	Web Services Manager Control initiates an SSL connection over HTTP with Oracle WSM Policy Manager. It sends updates to policies and retrieves and displays policy information in its user interface. Web Services Manager Control acts as a client for the SSL connection.	SSL over HTTP	"Securing Web Services Manager Control to Oracle WSM Policy Manager" on page 4-7
3	Web service clients can open connections to Oracle WSM Gateways over transport protocols such as HTTP, JMS, and MQ Series. Each of these protocols can be secured using SSL. The same connection is used to send the Web service request and to receive the response.	SSL over HTTP, JMS, or MQSeries	"Securing a Web Service Client to Oracle WSM Gateway" on page 4-8
4	The Oracle WSM Gateway can open connections to a Web service over transport protocols such as HTTP, JMS, and MQ Series. Each of these protocols can be secured using SSL. The same connection is used to send the Web service request and to receive the response.	SSL over HTTP, JMS, or MQSeries	"Securing Oracle WSM Gateway to a Web Service" on page 4-9
5	When an administrator connects to Web Services Manager Control, the Web browser can initiate an SSL connection over HTTP.	SSL over HTTP	"Securing Connections to Web Services Manager Control" on page 4-9
6	A Web service can be registered to an Oracle WSM Gateway by looking up a WSDL from a server hosting a WSDL, UDDI, or a WSIL file. Web Services Manager Control acts as a client to the server. This connection can be secured over SSL.	SSL over HTTP	"Registering a Service from a Secure WSDL" on page 4-9

Table 4–4 (Cont.) Description of Connections Between Oracle WSM Components

Figure Number	Description of the Connection Between Components	Supported Secure Transport Protocols	For Information on How to Configure Secure Connections
7	The Policy Enforcement Points can authenticate and authorize users against an LDAP Server. The gateway or agent acts as a client for the LDAPS connection.	SSL over LDAP	"Securing PEPs to an LDAP Server" on page 4-9
8	When users log in to Web Services Manager Control, they may be authenticated against an LDAP Server. The connection from Web Services Manager Control to the LDAP Server can be secured over SSL.	SSL over LDAP	"Securing Web Services Manager Control to an LDAP Server" on page 4-9
9	Web Services Manager Control sends data updates to Oracle WSM Database and retrieves data from Oracle WSM Database which it displays in its user interface over JDBC. SSL over JDBC connections to the Oracle WSM Database are not certified.	No secure transport protocols are supported.	Not Applicable
10	Oracle WSM Policy Manager sends information to and retrieves information from Oracle WSM Database over JDBC. SSL over JDBC connections to the Oracle WSM Database are not certified.	No secure transport protocols are supported.	Not Applicable
11	Oracle WSM Monitor sends monitoring data to and retrieves data from Oracle WSM Database over JDBC. SSL over JDBC connections to the Oracle WSM Database are not certified.	No secure transport protocols are supported.	Not Applicable
12	Web Services Manager Control retrieves and displays monitoring charts and reports from Oracle WSM Monitor over RMI. SSL over RMI connections to Oracle WSM Monitor are not certified.	No secure transport protocols are supported.	Not Applicable
13	Oracle WSM Agents and Oracle WSM Gateways send monitoring data to Oracle WSM Monitor over RMI. SSL over RMI connections to Oracle WSM Monitor are not certified.	No secure transport protocols are supported.	Not Applicable

Securing Connections to Oracle WSM Policy Manager

The following connections to Oracle WSM Policy Manager can be secured over SSL:

- Securing PEPs to Oracle WSM Policy Manager
- Securing Web Services Manager Control to Oracle WSM Policy Manager

Securing PEPs to Oracle WSM Policy Manager

Complete the following tasks to secure PEP connections to Oracle WSM Policy Manager:

- To SSL enable Oracle WSM Policy Manager, refer to "Securing OC4J Using SSL" on page 4-4.
- In addition, if connections to Oracle WSM Policy Manager are passed through Oracle HTTP Server, then you must also enable SSL for the `/policymanager/*` URL in your Oracle HTTP Server configuration. See *Oracle HTTP Server Administrator's Guide* for information on how to configure your Oracle HTTP Server.
- Configure PEPs to use a truststore by following the instructions in "Configuring the Truststore in SSL Connections" on page 4-2.

Securing Web Services Manager Control to Oracle WSM Policy Manager

Complete the following tasks to secure Web Services Manager Control connections to Oracle WSM Policy Manager:

- To SSL enable Oracle WSM Policy Manager, refer to "Securing OC4J Using SSL" on page 4-4.
- In addition, if connections to Oracle WSM Policy Manager are passed through Oracle HTTP Server, then you must also enable SSL for the `/policymanager/*` URL in your Oracle HTTP Server configuration. See *Oracle HTTP Server Administrator's Guide* for information on how to configure your Oracle HTTP Server.
- Configure the Web Services Manager Control by completing the following procedure:

  1. Edit the following properties in the *ORACLE_HOME*/owsm/config/ccore/ui-config-installer.properties file.

     ```
     ui.pm.server.httpScheme=https
     ui.pm.server.httpPort=SSL_port
     ```

     SSL_port is the port to which SSL connections are made.

  2. Redeploy the Web Services Manager Control application by executing the following command from the command line:

     ```
     wsmadmin deploy control
     ```

  3. Configure the Web Services Manager Control to use the truststore configuration. See "Configuring the Truststore in SSL Connections" on page 4-2 and follow the procedure for your deployment.

Securing Connections to and from Oracle WSM Gateway

Connections to and from Oracle WSM Gateway can be secured using SSL. The following sections describe how to create these secure connections:

- Securing a Web Service Client to Oracle WSM Gateway
- Securing Oracle WSM Gateway to a Web Service

Securing a Web Service Client to Oracle WSM Gateway

The following sections describe how to secure connections between a Web service client and Oracle WSM Gateway using SSL over HTTP, JMS, and MQ Series.

HTTPS

To SSL enable Oracle WSM Gateway, refer to "Securing OC4J Using SSL" on page 4-4.

In addition, if connections to Oracle WSM Gateway are passed through Oracle HTTP Server, then you must also enable SSL for the `/ gateway/*` URL in the HTTP Server configuration. See *Oracle HTTP Server Administrator's Guide* for information on how to configure your Oracle HTTP Server.

JMS and MQ Series for SSL

Secure the JMS and MQ Series protocols for SSL by following this procedure:

1. Edit the SSL properties in the *ORACLE_HOME*/owsm/config/gateway-config-installer.properties file.
2. Redeploy the Oracle WSM Gateway by executing the following command from the command line:

   ```
   wsmadmin deploy gateway
   ```

Securing Oracle WSM Gateway to a Web Service

Configure Oracle WSM Gateway to use a truststore by following the instructions in"Configuring the Truststore in SSL Connections" on page 4-2.

Securing Connections to Web Services Manager Control

To SSL enable Oracle WSM Gateway, refer to "Securing OC4J Using SSL" on page 4-4.

In addition, if connections to Web Services Manager Control are passed through Oracle HTTP Server, then you must also you must enable SSL for the `/ccore/*` URL in your Oracle HTTP Server configuration. See *Oracle HTTP Server Administrator's Guide* for information on how to configure your Oracle HTTP Server.

Registering a Service from a Secure WSDL

A Web service can be registered to a Oracle WSM Gateway by looking up a WSDL from a server hosting a WSDL, UDDI, or WSIL. Configure Web Services Manager Control to use a truststore by following the instructions in "Configuring the Truststore in SSL Connections" on page 4-2.

Securing Connections to an LDAP Server

The following connections to an LDAP Server can be secured over SSL:

- Securing PEPs to an LDAP Server
- Securing Web Services Manager Control to an LDAP Server

Securing PEPs to an LDAP Server

Configure PEPs to use a truststore by following the instructions in "Configuring the Truststore in SSL Connections" on page 4-2.

See *Oracle Web Services Manager Administrator's Guide* for information about policy pipelines and the LDAP authentication policy step.

Securing Web Services Manager Control to an LDAP Server

To use SSL to secure the connection to the LDAP Server, you must modify the properties in the *`ORACLE_HOME/`*`owsm/config/ccore/ui-config-installer.properties` file.

1. Modify the properties listed under the "UI authentication properties" property group in the `ui-config-installer.properties` file:

```
ui.authentication.provider=com.cfluent.accessprovider.ldap.BasicLdapAuthProvider

ui.authentication.provider.properties=

ldapHost=ldaphost|\

ldapPort=389|\

ldapDN=o=company,c=us

ldapSSLEnabled=true|\

ldapSSLPort=636|\

roleAttribute=uniqueMember
```

2. See "Configuring the Truststore in SSL Connections" on page 4-2 for instructions on how to configure the truststore.

Securing Web Services Manager Control to an Active Directory Server

To use SSL to secure the connection to the Active Directory Server, you must modify the properties in the *ORACLE_HOME*/owsm/config/ccore/ui-config-installer.properties file.

1. Modify the properties listed under the "UI authentication properties" property group in the `ui-config-installer.properties` file:

```
ui.authentication.provider=com.cfluent.accessprovider.ldap.ActiveDirectoryAuthProvider

ui.authentication.provider.properties=

ldapHost=ldaphost=139.185.17.7|

ldapPort=389|

ldapSSLEnabled=true|

ldapSSLPort=636|

ldapDN=dc=vanadium,dc=us,dc=oracle,dc=com|

roleAttribute=testgroup|

ldapDNSDomainName=vanadium.us.oracle.com|

ldapUidAttribute=sAMAccountName
```

Table 4–5 ui-config-installer.properties File Properties

Property	Description
`ui.authentication.provider`	Name of the class that authenticates users using Active Directory. This must be set to `com.cfluent.accessprovider.ldap.ActiveDirectoryAuthProvider`
`ldapHost`	Host name of the machine where the Active Directory Server is running.
`ldapPort`	Port on which the Active Directory Server listens for requests.
`ldapDN`	LDAP domain name
`ldapSSLEnabled`	Specifies whether SSL is enabled for the Active Directory Server.
`ldapSSLPort`	Port on which Active Directory listens for requests if SSL is enabled.
`roleAttribute`	LDAP attribute name that identifies the user in the LDAP group.
`ldapDNSDomainName`	Fully qualified domain name of the Active Directory Server.
`ldapUidAttribute`	Unique identifier used when searching through Active Directory Server. This must be set to `sAMAccountName`.

2. See "Configuring the Truststore in SSL Connections" on page 4-2 for instructions on how to configure the truststore.

Frequently Asked Questions

1. Why am I receiving a Host name verifier exception when SSL is enabled?

   The server-side SSL certificate's cn should match the host name of the component. For example, if the Policy Manager is running on `http://www.company.com`, then the server-side SSL certificate's dn should start with `cn=www.company.com`.

2. How do I create a new PKCS#12 keystore for testing using `openSSL`?

   The following example shows how to create a new PKCS#12 keystore with a self-signed certificate for the server side of a connection. The same process is used to create the corresponding client.p12 file.

   a. Create a new self-signed certificate, and a new private key to sign the certificate. Specify the time validity for the certificate with the `-days` option.

      ```
      openssl req -x509 -newkey rsa:1024 -days 365 -keyout
      server-privatekey.pem -out server-cert.pem
      ```

      You will be asked some questions about country, state, locality, and so on. These questions are asked for informational purposes. The exception to this is the Common Name (CN) field. The CN is used to represent the host name of the machine where the certificate resides. The SSL client may attempt to verify the host name with the CN field to see that they match. Make sure to set it to the host name that you have set for all the connection URLs in the various `config.xml` files (for example, `localhost`).

   b. Import the self-signed certificate and the associated private key into a PKCS#12 keystore.

      ```
      openssl pkcs12 -export -in server-cert.pem -inkey
      server-privatekey.pem -name testalias -out server.p12
      ```

   c. Delete any temporary files. Now that you have them stored in a PKCS#12 keystore, you do not need the `server-cert.pem` and `server-privatekey.pem` files.

      ```
      rm -f server-cert.pem server-privatekey.pem
      ```

3. Why am I getting the error message "HTTPS hostname wrong" when I try to connect to the Oracle WSM Policy Manager over SSL?

   The CN field of the server certificate does not match the host name to which the client is connecting. You need to ensure that the CN field matches the host name. An example error message (for localhost) is:

   ```
   java.io.IOException: HTTPS hostname wrong: should be
   <localhost>
   ```

   Note that if you are using a load balancer, the CN must match the load balancer host name.

4. How do I create a new JKS keystore for testing purposes using keytool?

   Enter the following command:

   ```
   keytool -genkey -keystore test.jks -storepass changeit
   ```

5

Deploying Oracle WSM Gateways

This chapter discusses how to deployOracle WSM Gateways, and includes the following sections:

- Oracle WSM Incoming and Outgoing Transport Protocols
- Registering Oracle Web Services Manager Gateways
- Configuring Incoming Transport Protocols
- Configuring the Incoming JMS Transport Protocol
- Configuring the Incoming MQ Transport Protocol

Oracle WSM Incoming and Outgoing Transport Protocols

The Oracle WSM Gateway receives message requests over an incoming protocol, and sends these requests over the outgoing protocol to the appropriate Web service. The incoming and outgoing protocols are independent of each other and configured separately. Within a single environment, you can mix or match the incoming and outgoing transport protocols.

For example, a client request can be made over HTTPS, and the Web service invocation can be made using Java Message Service (JMS). In this case, the Oracle WSM Gateway takes the incoming request and translates it to the outgoing transport protocol.

This chapter focuses on incoming transport protocols. See *Oracle Web Services Manager Administrator's Guide* for more information about outgoing transport protocols.

Before configuring the gateway to receive the incoming transport protocol, you need to register and deploy the Oracle Web Services Manager Gateway. The following section describes how to register and deploy a gateway.

Registering Oracle Web Services Manager Gateways

Before installing a Policy Enforcement Point (PEP), you need to register the PEP as a new component in the Oracle WSM Policy Manager. The registration process assigns an Oracle WSM component ID. The component ID defines the boundaries and responsibilities of the PEP. In the case of gateways, it contains information that teaches the gateway about your Oracle WSM environment.

Prerequisites and General Information about Registering Oracle WSM Gateways

Use Oracle Web Services Manager Control to register your Oracle WSM Gateway. Only users who are members of the group assigned to the Oracle WSM Domain

Administrator role can add or edit registration details for Oracle WSM PEPs: Oracle WSM Gateways, Oracle WSM Client or Server Agents.

Contact your Oracle WSM system administrator about obtaining the privileges required to access or update specific Oracle WSM components.

Refer to the *Oracle Web Services Manager Administrator's Guide* for more information about the Oracle WSM roles and permissions required to perform specific operations from the Web Services Manager Control.

Before you register your gateway, ensure that you have successfully completed the installation process for all server-side components.

To register a gateway

1. In the navigation pane of Web Services Manager Control, select **Policy Management**, then select the **Manage Policies** option.

   If any gateways or agents have been previously registered, the Web Services Manager Control displays a list of these registered components.

2. Click **Add New Component**.

   The Web Services Manager Control displays the Add New Component page.

3. Specify the following gateway properties:

   - Select **Gateway** from the list of Component Types.
   - Enter a meaningful name for the Component that describes its location or function. This is very important. Prior to registering new components, you may want to establish a convention for naming your components, so that you can easily identify them.
   - There is only one choice for Container Type – Oracle Web Services Manager.
   - In the Component URL field, specify the URL at which the gateway will be running. The syntax for specifying the URL is:

     `http://hostname:port/gateway`

     or

     `https://hostname:port/gateway`

     where *hostname* and *port* refer, respectively, to the host and port from which the gateway may be accessed.

     > **Note:** If you plan to load balance the gateway, then the host and port should refer to the externally visible host and port of the load balancer.

   - Enter the Component Groups to which you want to grant Modify or View permissions to the gateway.
     - Groups mapped to the Oracle WSM Component Administrator role may view and edit component details, and add, edit, or delete managed services and routing associated with the component. However, they cannot add, delete, or remove Oracle WSM components, or change any registration details of the components.
     - Groups mapped to the Oracle WSM Component Support role may only view component information and associated services. They are not

allowed to add, edit, or change details of the component, or any of its associated services.

See *Oracle Web Services Manager Administrator's Guide* for more information on roles and groups.

4. Click **Register**.

   Oracle WSM adds the component to the Oracle WSM system registry, assigns a unique ID to it, then displays the component ID.

5. Click **OK**.

   Oracle WSM displays the List of Components, including the newly added gateway and its component ID. Make a note of the component ID.

6. Edit the properties in the `ORACLE_HOME/owsm/config/gateway /gateway-config-installer.properties` file to configure incoming transport protocols.

7. Deploy the gateway using the `wsmadmin` command:

   ```
   wsmadmin deploy gateway
   ```

   See "deploy" on page C-12, for more information about the `deploy` command. If you make any other changes to the gateway, you must redeploy the component using the `wsmadmin deploy` command.

Configuring Gateway Component IDs

When you create and register an Oracle WSM Policy Enforcement Point (PEP), it is assigned a Component ID. Component IDs are unique identifiers that associate the PEP with a set of policies.

The component IDs start with the number C0003001. This number is incremented by 1 for each subsequent component that is created and registered. After a PEP is created, the component ID must be specified in the appropriate property file. (For gateways, this file is `gateway-config-installer.properties`. For agents, this file is `agent.properties.`) To make it easier to use Oracle WSM, by default, the Oracle WSM Gateway comes preconfigured in the `gateway-config-installer.properties` file with the component ID C0003001. This configuration assumes that the first component you create is an Oracle WSM Gateway.

There are two situations where this default configuration causes conflicts:

- The first PEP you create is an agent, not a gateway. In this case, the component ID C0003001 is assigned to the agent. If you subsequently create a gateway, it will be registered with the next component ID, in this example, C0003002. However, the component ID parameter in the `gateway-config-installer.properties` file is incorrectly assigned the number C0003001.
- If you have more than one Oracle WSM Gateway in your Oracle Web Services Manager environment and each gateway is assigned a different component ID, for example, C0003001 and C0003002. The component ID in the `gateway-config-installer.properties` file will be correct for the first gateway, but it will be incorrect for the second gateway.

In both of these situations, you must edit the properties file with the correct component ID.

To edit the component ID for an Oracle WSM Gateway

1. Open the following file in a text editor:
   ORACLE_HOME/owsm/config/gateway/gateway-config-installer.properties
2. Edit the value for the component ID in the parameter `gateway.component.id=`
3. Redeploy the gateway using the `wsmadmin deploy gateway` operation.

   See "deploy" on page C-13 for more information on this command. See Table 6, "Installing Oracle WSM Agents" for more information on editing the component ID for Oracle WSM Agents.

Configuring Incoming Transport Protocols

Gateways support the reception of both SOAP messages and standard XML message requests over JMS, MQ, and HTTPS. The *ORACLE_HOME*/owsm/config/gateway/gateway-config-installer.properties file for each gateway specifies how that gateway listens for messages in these different formats. There are sections in the `gateway-config-installer.properties` file for each of the protocols (HTTPS, JMS, and MQ), each associated with a pair of processable message types (SOAP or standard XML) (Table 5–1).

Table 5–1 Gateway Transport Protocols

Parameter	Description
JMS	Listens for SOAP messages arriving at a JMS queue.
JMSXML	Listens for XML (non-SOAP) messages sent to a JMS queue.
MQ	Listens on an MQ queue for SOAP messages.
MQXML	Listen on a MQ queue for XML (non-SOAP) messages.
HTTP	Listens on a HTTP port for SOAP or XML messages. (The gateway is enabled by default.)

The sections that follow describe the parameters for each of the following protocols:

- JMS – See Table 5–2 on page 5-5 for more information.
- JMS XML – See Table 5–3 on page 5-7 for more information.
- MQ – See Table 5–4 on page 5-9 for more information.
- MQ XML – See Table 5–5 on page 5-10 for more information.

Configuring the Incoming JMS Transport Protocol

JMS requires only one handshake for each session, rather than requiring a secure handshake for every message. JMS facilitates transport without the latency associated with HTTPS. You can set up JMS to transport either SOAP or XML messages.

The number of the JMS Servers that you need to deploy and configure for your system depends on the number of deployed Oracle WSM Monitors in your Oracle WSM environment. A single, centralized Oracle WSM Monitor requires only a single, central JMS server. Federated Oracle WSM Monitors may require separate messaging systems, and as a result, multiple JMS servers.

The `gateway-config-installer.properties` file has two JMS sections that specify properties required for starting the JMS request handler. The first section, JMS properties, contains the properties to use JMS to process SOAP messages. The second section, JMS XML, contains the properties to use JMS to process standard XML messages.

After you edit the `gateway configuration properties` file, you must redeploy the gateway using the `wsmadmin deploy` command for the changes to take effect.

Note: Oracle Web Services Manager supports JMS queues to enable point-to-point functionality.

Enabling Oracle Web Services Manager to use JMS

To enable JMS in Oracle Web Services Manager, you must complete the following tasks:

- If you are using Tibco JMS, copy the `tibjms.jar` and `tibcrypt.jar` files to the correct location in your Oracle WSM installation, for example, to *`ORACLE_HOME`*`/10.1.3.1/OracleAS_1/owsm/lib/custom`. Unless you perform this task, the JMS feature will not work.

  Note: You do not need to copy the Java Archive (JAR) files if you are using Oracle JMS.

- Configure JMS listeners on a gateway.
- Register and configure a JMS service on a gateway (using JMSMessengerStep). See the *Oracle Web Services Manager Administrator's Guide* for more information.

To test whether you can connect over the JMS protocol, complete the following tasks:

- Test the configured JMS service using the Test Engine tool in the Oracle Web Services Manager Control.
- Enable the sending of Direct Internet Message Encapsulation (DIME) messages.

Table 5–2 lists the properties you can set for JMS transport message request handling.

Table 5–2 JMS Properties

Property	Description
`gateway.listener.jms.Enabled`	Supported values are `true` or `false`. Starts the JMS request handler when set to `true`. (The remainder of the parameters in this table are relevant only when this parameter is set to `true`).
`gateway.listener.jms.useJndi`	Supported values are `true` or `false`. Enables or disables JNDI for retrieving queue information.
`gateway.listener.jms.sslEnabled`	Supported values are `true` or `false`. If set to `true`, enables JMS communication over SSL.
`gateway.listener.jms.queueName`	The name of the queue where the JMS request handler listens. When JNDI is enabled, this is the JNDI name of the Queue.

Table 5–2 (Cont.) JMS Properties

Property	Description
`gateway.listener.jms.retryInterval`	The time in seconds that the gateway waits before checking JMS server availability. If the server is not available, the gateway pings and checks the server availability at the next retry interval.
`gateway.listener.jms.defaultServiceId`	The default service ID used when the destination service ID is not specified.
`gateway.listener.jms.env.connectionFactory`	The connection factory class name used to get the queue connection. JNDI looks up the connection factory to fetch the actual queue.
`gateway.listener.jms.env.username`	The user name that enables you to connect to the JMS server and fetch the queue.
`gateway.listener.jms.env.password`	The password for connecting to the JMS server. Use the encryption tool to ensure that your password is sent in an encrypted format.
`gateway.listener.jms.env.url`	The URL used to connect to the JMS server.
`gateway.listener.jms.ssl.sslVendor`	Supported values are J2SE and entrust6 for Tibco JMS. The vendor that provides the SSL certificates.
`gateway.listener.jms.ssl.sslHostName`	Name of the JMS server, expected in the JMS certificate.
`gateway.listener.jms.ssl.clientIdentity`	Client identity certificate or the path to the certificate file.
`gateway.listener.jms.ssl.clientIdentityPassword`	Password to decrypt the provided identity file. Use the encryption tool to ensure that your password is sent in an encrypted format.
`gateway.listener.jms.ssl.trustCerts.cert`	Trust certificate of the server or the path to the certificate file.
`gateway.listener.jms.jndi.providerUrl`	The jndi.providerURL that gets the jndi context.
`gateway.listener.jms.jndi.providerCtxFactory`	The jndi.providerContextFactory class used to get the JNDI context.
`gateway.listener.jms.jndi.urlPackagePrefix`	Value used to initialize the JNDI context.
`gateway.listener.jms.jndi.username`	User name for the JNDI lookup.
`gateway.listener.jms.jndi.password`	Password for the JNDI lookup. Use the encryption tool to ensure that your password is sent in an encrypted format.

Table 5–3 lists the properties you can set for JMS XML transport message request handling.

Table 5–3 JMS XML Properties

Property	Description
`gateway.listener.jmsxml.enabled=false`	Supported values are `true` or `false`. Starts the JMS XML request handler when set to `true`. (The remainder of the parameters in this table are relevant only when this parameter is set to `true`).
`gateway.listener.jmsxml.useJndi`	Supported values are `true` or `false`. Enables or disables JNDI for retrieving queue information.
`gateway.listener.jmsxml.sslEnabled`	Supported values are `true` or `false`. If set to `true`, enables JMSXML communication over SSL.
`gateway.listener.jmsxml.queueName`	The name of the queue where the JMS XML request handler listens. When JNDI is enabled, this is the JNDI name of the Queue.
`gateway.listener.jmsxml.retryInterval`	The time in seconds that the gateway waits before checking JMS server availability. If the server is not available, the gateway pings and checks the server availability at the next retry interval.
`gateway.listener.jmsxml.defaultServiceId`	The default service ID used when the destination service ID is not specified.
`gateway.listener.jmsxml.env.connectionFactory`	The connection factory class name used to get the queue connection. JNDI looks up the connection factory to fetch the actual queue.
`gateway.listener.jmsxml.env.username`	The user name that enables you to connect to the JMS server and fetch the queue.
`gateway.listener.jmsxml.env.password`	The password for connecting to the JMS server. Use the encryption tool to ensure that your password is sent in an encrypted format.
`gateway.listener.jmsxml.env.url`	The URL used to connect to the JMS server.
`gateway.listener.jmsxml.ssl.sslVendor`	Supported values are J2SE and entrust6 for Tibco JMS. The vendor that provides the SSL certificates.
`gateway.listener.jmsxml.ssl.sslHostName`	Name of the JMS server, expected in the JMS certificate.
`gateway.listener.jmsxml.ssl.clientIdentity`	Client identity certificate or the path to the certificate file.

Table 5–3 (Cont.) JMS XML Properties

Property	Description
`gateway.listener.jmsxml.ssl.clientPassword`	Password to decrypt the provided identity file. Use the encryption tool to ensure that your password is sent in an encrypted format.
`gateway.listener.jmsxml.ssl.trustCerts.cert`	Trust certificate of the server or the path to the certificate file.
`gateway.listener.jmsxml.jndi.providerUrl`	The jndi.providerURL that gets the jndi context.
`gateway.listener.jmsxml.jndi.providerContextFactory`	The jndi.providerContextFactory class used to get the JNDI context.
`gateway.listener.jmsxml.jndi.urlPackagePrefix`	Value used to initialize the JNDI context.
`gateway.listener.jmsxml.jndi.username`	User name for the JNDI lookup.
`gateway.listener.jmsxml.jndi.password`	Password for the JNDI lookup. Use the encryption tool to ensure that your password is sent in an encrypted format.

Configuring the Incoming MQ Transport Protocol

MQ series allows the use of the following Java implementations:

- **MQSeries base classes for Java (MQ base Java) –** Allows a program written in the Java programming language to connect to MQSeries as an MQSeries client. This involves direct connection to an MQSeries server, and enables Java applets, applications, and servlets to issue calls and queries to MQSeries.
- **MQSeries classes for Java Message Service (MQ JMS) –** A set of Java classes that implement Sun Microsystem's Java Message Service (JMS) interfaces to enable JMS programs to access MQSeries systems. Both the point-to-point and publish-and-subscribe models of JMS are supported.

Enabling Oracle Web Services Manager to use MQ

Oracle Web Services Manager uses MQSeries classes for Java (MQ base Java).

To enable MQ in Oracle Web Services Manager, complete the following tasks:

- Acquire an MQ license. The MQ license is not bundled with Oracle WSM, therefore, you must acquire the correct license. Then, you must copy the `com.ibm.mq.jar` file to the correct location in your Oracle WSM installation, for example, to *`ORACLE_HOME`*`/owsm/lib/custom`. Unless you perform this task, the MQ feature will not work.
- Configure MQ listeners on a gateway.
- Register and configure an MQ service on a gateway (using MQMessengerStep). See *Oracle Web Services Manager Administrator's Guide* for more information.

To test whether you can connect over the MQ protocol, complete the following tasks:

- Test the configured MQ service using the Test Engine tool in the Oracle Web Services Manager Control.

- Enable the sending of DIME messages.

Setting up MQ Listeners in a Gateway

MQ transport protocols listen for a client request to be processed on a configured MQ queue.

The `gateway-config-installer.properties` file has two MQ sections that specify properties required for starting the MQ request handler. The first section, MQ properties, contains the properties to use MQ to process SOAP messages. The second section, MQ XML, contains the properties to use MQ to process standard XML messages. After editing the gateway configuration file, you must redeploy the gateway using the `wsmadmin deploy` command for the changes to take effect.

Table 5–4 lists the properties you can set to configure MQ transport message request handling.

Table 5–4 MQ Properties

Property	Description
`gateway.listener.mq.enabled`	The default value is false. Starts the MQ request handler when set to `true`. (The rest of the parameters in this table are relevant only when the parameter on the left is set to `true`).
`gateway.listener.mq.managerName`	Name of the queue manager to which the application needs to connect to access the queues configured on the MQSeries server.
`gateway.listener.mq.hostName`	Valid name or IP address of the host where MQSeries server is running.
`gateway.listener.mq.port`	Valid port number of the MQSeries server listener.
`gateway.listener.mq.username`	User name by which the application can connect to the MQSeries Server.
`gateway.listener.mq.password`	Password for the user by which the application can connect to MQSeries.
`gateway.listener.mq.channelName`	Name of the channel which connects an application to a queue manager on a server machine for the transfer of MQ calls and responses.
`gateway.listener.mq.queueName`	Name of the queue where the MQ Listener waits for request messages.
`gateway.listener.mq.retryInterval`	Interval between retries for retrieving messages from the MQ server.
`gateway.listener.mq.defaultServiceId`	Service to which messages will be forwarded by the MQ listener if the incoming message does not specify a destination service.

Table 5–5 lists the properties you can set to configure MQ XML transport message request handling.

Table 5–5 MQ XML Properties

Property	Description
`gateway.listener.mqxml.enabled`	The default value is false. Starts the MQ XML request handler when set to `true`. (The remainder of the parameters in this table are relevant only when this parameter is set to `true`). To enable IBM MQ, set the value to true.
`gateway.listener.mqxml.managerName`	Name of the queue manager to which the application needs to connect to access the queues configured on the MQSeries server.
`gateway.listener.mqxml.hostName`	Valid name or IP address of the host where MQSeries server is running.
`gateway.listener.mqxml.port`	Valid port number of the MQSeries server listener.
`gateway.listener.mqxml.username`	User name by which the application can connect to the MQSeries Server.
`gateway.listener.mqxml.password`	Password for the user by which the application can connect to MQSeries.
`gateway.listener.mqxml.channelName`	Name of the channel which connects an application to a queue manager on a server machine for the transfer of MQ calls and responses.
`gateway.listener.mqxml.queueName`	Name of the queue where the MQ Listener waits for request messages.
`gateway.listener.mqxml.retryInterval`	Interval between retries for retrieving messages from the MQ server.
`gateway.listener.mqxml.defaultServiceId`	Service to which messages will be forwarded by the MQ listener if the incoming message does not specify a destination service.

6

Installing Oracle WSM Agents

This chapter describes how to install Oracle Web Services Manager (Oracle WSM) Agents and includes the following sections:

Overview of Oracle WSM Agents

Oracle WSM Agents are Policy Enforcement Points (PEP) that execute policies in the same process space as the target Web service or the Web service client they protect. Oracle WSM Agents provide end-to-end security from the client request to the target Web service. They check whether they are configured to communicate with Oracle WSM Policy Manager. If they are, they retrieve the security information from Oracle WSM Policy Manager. If Oracle WSM Policy Manager is disabled, then the Oracle WSM Agents look for a local file that contains the security policies to be executed.

There are two main types of deployments using Oracle WSM Agents:

- **Oracle WSM Client and Server Agents** – In this deployment (Figure 6–1), the client agent intercepts the Web service request and applies a policy. The policy can be configured so that the client agent inserts security artifacts into the SOAP header, for example, a signature, an encryption, or tokens for authentication. The server agent verifies these security artifacts before it passes the Web service request to the Web service. Similarly, the Web service response can be secured by the server agent. The server agent can insert security artifacts into the SOAP header which are verified by the client agent before the client agent passes the response to the client.

Figure 6–1 Deployment with Oracle WSM Client and Server Agents

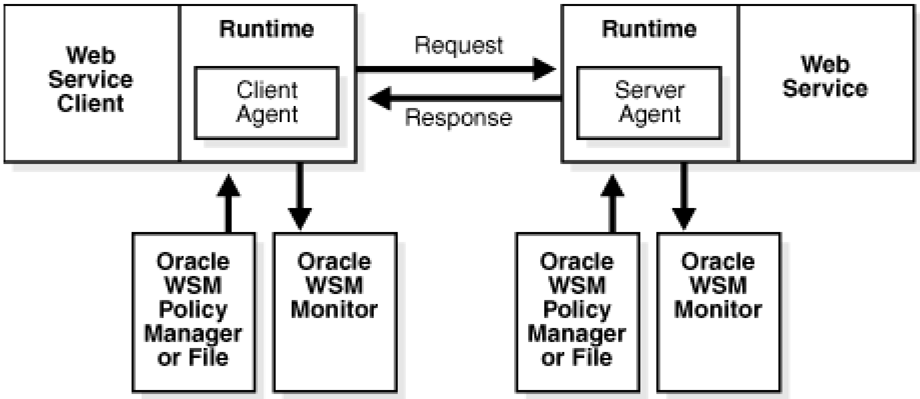


Figure 6–1 shows an Oracle WSM Agent at both ends of the transaction. However, it is not a requirement that both agents must be Oracle WSM Agents. In typical business-to-business transactions, one or the other agent is not an Oracle agent.

- **Oracle WSM Gateway with Oracle WSMAgents** – In this deployment (Figure 6–2), the Oracle WSM Gateway is set up in the DMZ. While the Oracle WSM Gateway can provide security for Web requests coming from the client, there is a potential security hole between the Oracle WSM Gateway and the Web service. A server agent can be used to prevent such security attacks. This is known as last-mile security.

Figure 6–2 Oracle WSM Gateway with Oracle WSM Agents

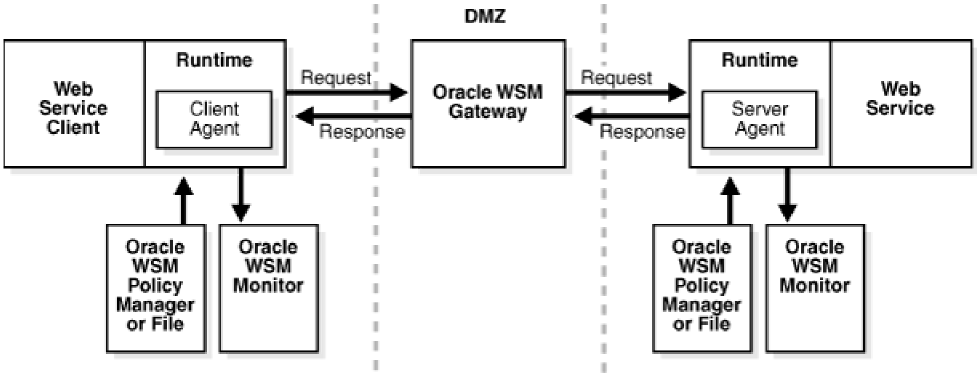
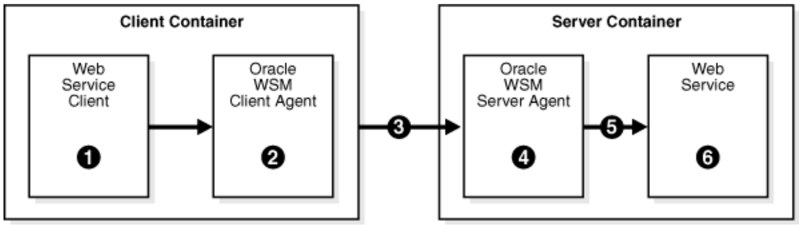


Figure 6–3 illustrates how Oracle WSM client agents and server agents are used. In actual business-to-business situations, the client or the server may not have Oracle WSM. Because Oracle WSM implements industry standards such as WS-Security, Oracle WSM Agents are compatible with other standards-compliant security systems.

Figure 6–3 Using Oracle WSM Client and Server Agents to Enforce Policies



The numbers in Figure 6–3 correspond to the following steps that describe how a client request to a server application is intercepted by the Oracle WSM Agents:

1. The Web service client makes a request to the Web service.
2. The Oracle WSM Client Agent intercepts the outgoing request and enforces the policies in its Request pipeline.
3. After the policy steps have been successfully executed by the Oracle WSM Client Agent, the request is sent to the Web service.
4. The Oracle WSM Server Agent intercepts the incoming request and enforces the policies defined in its Request pipeline.
5. After the policy steps have been successfully executed by the Oracle WSM Server Agent, the request is passed on to the Web service.
6. The Web service processes the request.

Response pipelines may be defined for the Oracle WSM Server Agent or the Oracle WSM Client Agent, or for both agents. If this is the case, when the Web service sends its response back to the Web service client, the outgoing message is first intercepted by the Oracle WSM Server Agent. If policies are defined in the Response pipeline, the Oracle WSM Server Agent executes these policies. The message is then intercepted by the Oracle WSM Client Agent before the response is finally passed back to the Web service client application.

Installing Oracle WSM Agents Overview

The following is an overview of the installation steps for the Oracle WSM agents:

1. Use Oracle Enterprise Manager 10*g* Web Services Manager Control Console (Web Services Manager Control Console) to register the Oracle WSM client or server agent component with the Oracle WSM Policy Manager. Policies can be defined at the time an Oracle WSM agent is registered, or they can be defined at a later time.
2. Execute the WSMADMIN tool to install the registered Oracle WSM agent.
3. Configure and assemble the agent into the Web service or the Web service client.

   For OC4J clients, this step is non-intrusive because the Web service or the Web service client is not modified. The Oracle WSM agent is part of the OC4J runtime or the client runtime.

The sections that follow describe how to install each type of Oracle WSM Agent.

Installing Server Agents

This section describes how to install server agents for the following Web services on OC4J:

- Oracle Web service
- Apache Axis 1.1 and 1.4 (AXIS) Web service

Installing a Server Agent for an Oracle Web Service

Complete the following steps to install a server agent for an Oracle Web service:

1. Add a server agent.
2. Define a policy for the server agent.

3. Install the server agent.
4. Configure the Web service deployment descriptor.

The sections that follow describe each of these steps in detail.

To add a server agent

1. Use the Web Services Manager Control Console to register a server agent, and select the following values from the drop-down lists.
   a. **Component type** – Server Agent
   b. **Container type** – OC4J
2. Select **Register**. This generates a component ID. Make a note of this component ID because you will use this ID when you install the server agent.

To define a policy for the server agent

From Web Services Manager Control, configure the policy you want to associate with the agent. See Chapter 5, "Oracle Web Services Manager Policy Management," in *Oracle Web Services Manager Administrator's Guide* for more information on how to define a policy for an agent.

To install the server agent

Check that the Oracle Application Server is running before you install the server agent.

1. Edit the attributes in the `ORACLE_HOME/owsm/bin/agent.properties` file with the following values:
   - `agent.componentType` – OC4JServerInterceptor
   - `agent.containerType` – OC4J
   - `agent.containerVersion` – 10.1.3
   - `agent.component.id` – Enter the component ID that is generated when the agent is created and registered using Web Services Manager Control.
2. Save the changes to the file.

   > **Note:** Keep a copy of the `agent.properties` file so that you can reuse the configurations if you need to reinstall the agent at a later time. If you create multiple agents with different configurations, keep copies of the different `agent.properties` files for each configuration.

3. The following properties should be automatically populated. Verify that the values are correct for these properties:
   - `oc4j.home`
   - `oc4j.j2ee.home`
   - `external.oc4j.port`
   - `external.oc4j.adminPort`
   - `external.oc4j.adminID`

   Refer to "The agent.properties File" on page 6-15 for information on these properties.

4. Install the agent by executing the `wsmadmin installAgent` command.

> **Note:** Verify that the Oracle Application Server, on which the server agent is to be installed, is up and running before you execute this command.

See "installAgent" on page C-22 in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information about this command.

To configure the Web service deployment descriptor

The following procedure is performed on the Oracle Application Server where the application that contains the Web service that is to be protected resides.

1. From the Oracle Enterprise Manager 10*g* Application Server Control Console, select the application that contains the Web service to be protected with the Oracle WSM server agent.
2. Click the **Web Services** tab.
3. Click the Web service to be protected, then click the **Administration** tab.
4. Click **Enable/Disable Features.**
5. Move the Web Services Agent from the Available Features box to the Enabled Features box, and click **OK.**
6. Click **Edit Configuration** icon for the Web Services Agent.
7. In the Configuration Directory box, enter the component ID for the server agent. The value you enter here should match the value you entered for the `agent.component.id` property in the `agent.properties` file.
8. Click **Save.**

Installing a Server Agent for an AXIS 1.1 or 1.4 Web Service

Before you install the server agent, make sure that the AXIS Web application has a `server-config.wsdd` file that describes the Web services deployed by the application. Refer to AXIS documentation for more information about this file.

Complete the following steps to install a server agent for an AXIS Web service:

1. Add a server agent.
2. Define a policy for the server agent.
3. Install the server agent.
4. Redeploy the Web service application.

The sections that follow describe each of these steps in detail.

To add a server agent

1. Use the Web Services Manager Control to create a server agent, and select the following values from the drop-down lists.
   a. **Component type** – Server Agent
   b. **Container type** – AXIS
2. Select **Register**. This generates a component ID. Make a note of this component ID because you will use this ID when you install the server agent.

To define a policy for the server agent

From Web Services Manager Control, configure the policy you want to associate with the agent. See Chapter 5, "Oracle Web Services Manager Policy Management," in *Oracle Web Services Manager Administrator's Guide* for more information on how to define a policy for an agent.

To install the server agent

1. Edit the attributes in the `agent.properties` file in the *`ORACLE_HOME`*`/owsm/bin` directory with the following values:
   - `agent.componentType` – serveragent
   - `agent.containerType` – AXIS
   - `agent.containerVersion` – Enter the version of AXIS. The valid values are *1.1* or *1.4*.
   - `agent.component.id` – Enter the component ID that is generated when the agent is created and registered using Web Services Manager Control.
2. If the Web service application is packaged as a Web Archive (WAR) file, then edit the `webservice.application.input` property in the `agent.properties` file. Provide the full path and name of the WAR file.
3. If the Web service application is packaged as an EAR file, then edit the following properties in the `agent.properties` file:
   - `webservice.application.input` – Enter the full path and name of the EAR file.
   - `webservice.application.webapp.name` – Uncomment and enter the WAR file name, for example, `hellows-web.war`.
   - `webservice.application.contexturi` – Provide the context root, for example, `/hellows.`

   See Table 6–1, " agent.properties File" for more information on the `agent.properties` file.

   > **Note:** Keep a copy of the `agent.properties` file so that you can reuse the configurations if you need to reinstall the agent at a later time. If you create multiple agents with different configurations, keep copies of the different `agent.properties` files for each configuration.

4. Execute the `wsmadmin installAgent` command.

   See "installAgent" on page C-22 in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information about this command.

Redeploying the Web Service Application

In order for the server agent to work, the application must be redeployed so that the changes take effect. See *Oracle Containers for J2EE Deployment Guide* for information about deploying Web applications.

Installing Client Agents

This section describes how to install client agents for the following Web service clients:

- **Oracle J2SE Client –** This is a standalone Java application, typically a Swing application on the end user's computer, that accesses a Web service. Client agents for J2SE clients filter outgoing and incoming messages between the Java application and the Web service.
- **Oracle J2EE Client –** This is a J2EE module, hosted in an application server, that accesses a Web service. The client agent filters outgoing and incoming messages between the J2EE module and the Web service. This client may be a servlet, JavaServer Page (JSP), or Enterprise JavaBean (EJB). Or it can be a J2EE application client that is a standalone application launched from the command line; typically, this J2EE application client accesses EJB programs running on a J2EE application server.
- **AXIS Client –** This is a Web service client written using the AXIS framework. Oracle WSM 10.1.3.3 only supports AXIS 1.1 and 1.4 J2EE clients.
- **Client Agent for BPEL or ESB Process –** This is a BPEL or ESB process that calls an external Web service. The client agent filters outgoing and incoming messages between the BPEL or ESB process and the Web service

> **Note:** Before you install the client agent, test the Web service client and verify that the client is getting the intended result from the Web service. Refer to the following documents for more information on developing Web service clients:
>
> - *Oracle Application Server Web Services Developer's Guide*
> - *Oracle Application Server Advanced Web Services Developer's Guide*

Installing a Client Agent for an Oracle J2SE Client

Complete the following steps to install a client agent for an Oracle J2SE client:

1. Add a client agent.
2. Define a policy for the client agent.
3. Install the client agent.
4. Configure the client agent with the Oracle J2SE Web service client.
5. Set up the runtime environment.

The sections that follow describe each of these steps in detail.

To add a client agent

1. Use the Web Services Manager Control to create a client agent, and select the following values from the drop-down lists.
   a. **Component type** – Client Agent
   b. **Container type** – OC4J
2. Select **Register.** This generates a component ID. Make a note of this component ID because you will use this ID when you install the client agent.

To define a policy for the client agent

Configure the policy you want to associate with the agent. See Chapter 5, "Oracle Web Services Manager Policy Management," in *Oracle Web Services Manager Administrator's Guide* for more information on how to define a policy for an agent.

To install the client agent

> **Note:** Oracle WSM must be installed on the machine where you are installing the client agent. The following procedure assumes that Oracle WSM is installed in the location specified by *ORACLE_HOME*.

1. Edit the following attributes in the *ORACLE_HOME*/owsm/bin/agent.properties file:
   - agent.componentType – OC4JClientInterceptor
   - agent.containerType – OC4J
   - agent.containerVersion – 10.1.3
   - client.home – Enter the full path to the location of the client, for example, c:\oracle\client.
   - agent.component.id – Enter the component ID that was generated when the agent is created and registered using Web Services Manager Control.

   > **Note:** Keep a copy of the agent.properties file so that you can reuse the configurations if you need to reinstall the agent at a later time. If you create multiple agents with different configurations, keep copies of the different agent.properties files for each configuration.

2. From the command prompt, install the client agent using the wsmadmin installAgent command.

   This installs the client agent at the location specified in the client.home attribute in the agent.properties file. For example, if client.home is specified as c:\oracle\client and assuming that the component ID of the agent is C0003006, then the following directory is created:

   ```
   C:\oracle\client\owsm\config\interceptors\C0003006
   ```

   Several configuration files and subdirectories are copied to this location.

   See "installAgent" on page C-22 in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information about this command.

Configuring the Client Agent with the Oracle J2SE Web Service Client

When the Web service client is created and compiled, there are several Java artifacts. Specifically, there is a Java file with a name in the following form: *package*.runtime.*binding*_Stub.java

For example: org.myorg.myclient.runtime.HttpSoap11Binding_Stub.java

A configuration file is generated with the same name as the .java file except it has an .xml extension. The name of the configuration file is in the following form: *package*.runtime.*binding*_Stub.xml

For example:

```
org.myorg.myclient.runtime.HttpSoap11Binding_Stub.xml
```

If this file is not automatically generated, create the file using the oracle-webservices-client-10_0.xsd schema. The configuration file must be present in the class path of the client just as the `org.myorg.myclient.runtime.HttpSoap11Binding_Stub.xml` would be.

Add the `<runtime enabled="owsm" ...</runtime>` tag into your *`package`*`.runtime.`*`binding`*`_Stub.xml` file. The following lines are from the example `org.myorg.myclient.runtime.HttpSoap11Binding_Stub.xml` configuration file.

```
<oracle-webservice-clients>
<webservice-client>
<port-info>
<runtime enabled="owsm">
<owsm
init-home="C:/oracle/client/owsm/config/interceptors/C0003005"/>
</runtime>
</port-info>
</webservice-client>
</oracle-webservice-clients>
```

The value specified for the `init-home` attribute of the `owsm` element must be the location to which the agent configuration was copied in the "To install the client agent" procedure on page 6-8.

Setting Up the Runtime Environment

Before you run the Web service client, add the following JAR files to the classpath:

- All JAR files in the following location: *`client.home`*`/owsm/lib/extlib`. The variable *`client.home`* is the directory into which the Web service client is installed.
- *`ORACLE_HOME`*`/owsm/lib/cfluent-log4.j.jar`
- JDBC JAR File. For example, for the Basic standalone installation or the Basic SOA installation, this location is: *`ORACLE_HOME`*`/Mobile/Sdk/BIN/OLITE40.JAR`. If you installed the Advanced standalone or Advanced SOA option, refer to the *Oracle Database Administrator's Guide* to find the location of the JDBC JAR file.
- *`ORACLE_HOME`*`/jlib/orai18n.jar` and *`ORACLE_HOME`*`/jlib/ojmisc.jar`.

Installing a Client Agent for an Oracle J2EE Client

Complete the following steps to install a client agent for an Oracle J2EE client:

1. Add a client agent.
2. Define a policy for the client agent.
3. Install the client agent.
4. Configure the client agent with the Oracle J2EE Web service client.
5. Redeploy the Web service client (EJB and Servlet clients only).
6. Set up the runtime environment (J2EE application clients only).

The sections that follow describe each of these steps in detail.

To add a client agent

1. Use the Web Services Manager Control to create a client agent, and select the following values from the drop-down lists.

   a. **Component type** – Client Agent

   b. **Container type** – OC4J

2. Select **Register.** This generates a component ID. Make a note of this component ID because you will use this ID when you install the client agent.

To define a policy for the client agent

Create the policy you want to associate with the agent. See Chapter 5, "Oracle Web Services Manager Policy Management," in *Oracle Web Services Manager Administrator's Guide* for more information on how to define a policy for an agent.

To install the client agent

> **Note:** Oracle WSM must be installed on the machine where you are installing the client agent. The following procedure assumes that Oracle WSM is installed in the location specified by *`ORACLE_HOME`*.

1. Edit the following attributes in the *`ORACLE_HOME`*`/owsm/bin/agent.properties` file:
   - `agent.componentType` – OC4JClientInterceptor
   - `agent.containerType` – OC4J
   - `agent.containerVersion` – 10.1.3
   - `client.home` – Enter the path where the client agent configuration will be installed. Oracle recommends that you use *`ORACLE_HOME`* for `client.home`.
   - `agent.component.id` – Enter the component ID that was generated when the agent is created and registered using Web Services Manager Control.

   > **Note:** Keep a copy of the `agent.properties` file so that you can reuse the configurations if you need to reinstall the agent at a later time. If you create multiple agents with different configurations, keep copies of the different `agent.properties` files for each configuration.

2. From the command prompt, install the client agent using the `wsmadmin installAgent` command.

   This installs the client agent at the location specified in the `client.home` attribute in the `agent.properties` file. For example, if `client.home` is specified as `c:\oracle\client` and assuming that the component ID of the agent is C0003006, then the following directory is created:

   ```
   C:\oracle\client\owsm\config\interceptors\C0003006
   ```

   Several configuration files and subdirectories are copied to this location.

   See "installAgent" on page C-22 in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information about the `wsmadmin installAgent` command.

Configuring the Client Agent with the Oracle J2EE Web Service Client

Identify the location of the client agent configuration to the Oracle J2EE Web service client. Edit the properties of the appropriate J2EE client deployment descriptor file:

- Servlets and JSPs – Edit the `WEB-INF/orion-web.xml` file in the WAR archive.
- EJB – Edit the `META-INF/orion-ejb-jar.xml` file.
- Application Clients – Edit the `META-INF/orion-application-client.xml` file in the application client's JAR file.

Add the `<runtime enabled="owsm" ...</runtime>` tag into the appropriate J2EE client deployment descriptor file. The following is an example of how a client deployment descriptor is configured with the location of the client agent configuration.

```
...
<service-ref-mapping name="...">
<port-info>
<wsdl-port namespaceURI="..." localport="..."/>
<runtime enabled="owsm">
<owsn
init-home="C:/oracle/client/owsm/config/interceptors/C0003006"/>
</runtime>
...
</port-info>
</service-ref-mapping>
```

Redeploying the Web Service Client

For EJB and Servlet clients, you must redeploy the Web service client after you configure the client agent. This step assumes that the client agent is installed in *ORACLE_HOME*.

Setting Up the Runtime Environment

> **Note:** This step assumes that the client agent is installed in *ORACLE_HOME*.

For J2EE application clients, add the following JAR files to the classpath of the Web service client:

- All JAR files in the following location: *ORACLE_HOME*`/owsm/lib/extlib`. The variable *`client.home`* is the directory into which the Web service client is installed.
- *ORACLE_HOME*`/owsm/lib/cfluent-log4.j.jar`
- JDBC JAR File. For example, for the Basic standalone installation or the Basic SOA installation, this location is: *ORACLE_HOME*`/Mobile/Sdk/BIN/OLITE40.JAR`. If you installed the Advanced standalone or Advanced SOA option, refer to the *Oracle Database Administrator's Guide* to find the location of the JDBC JAR file.
- *ORACLE_HOME*`\jlib\orai18n.jar` and *ORACLE_HOME*`\jlib\ojmisc.jar`.

Installing a Client Agent for an AXIS 1.1 or 1.4 J2EE Client

Complete the following steps to install a client agent for an AXIS 1.1 or 1.4 J2EE client:

1. Add a client agent.
2. Define a policy for the client agent.
3. Install the client agent.

4. Redeploy the Web service client.

The sections that follow describe each of these steps in detail.

To add a client agent

1. Use the Web Services Manager Control to create a client agent, and select the following values from the drop-down lists.

   a. **Component type** – Client Agent

   b. **Container type** – AXIS

2. Select **Register.** This generates a component ID. Make a note of this component ID because you will use this ID when you install the client agent.

To define a policy for the client agent

Configure the policy you want to associate with the agent. See Chapter 5, "Oracle Web Services Manager Policy Management," in *Oracle Web Services Manager Administrator's Guide* for more information on how to define a policy for an agent.

To install the client agent

1. Edit the following attributes in the `*ORACLE_HOME*/owsm/bin/agent.properties` file:

   - `agent.componentType` – clientagent
   - `agent.containerType` – AXIS
   - `agent.containerVersion` – Enter the version of AXIS. The valid values are *1.1* and *1.4*.
   - `agent.component.id` – Enter the component ID that was generated when the agent is created and registered using Web Services Manager Control.

2. If the Web service application is packaged as a Web Archive (WAR) file, then edit the `webservice.application.input` property in the `agent.properties` file. Provide the full path and name of the WAR file.

3. If the Web service client is packaged as an EAR file, then edit the following properties in the `agent.properties` file:

   - `webservice.application.input` – Enter the full path and name of the EAR file.
   - `webservice.application.webapp.name` – Uncomment and enter the WAR file name, for example, `hellows-web.war`.
   - `webservice.application.contexturi` – Provide the context root, for example, `/hellows`.

   See Table 6–1, " agent.properties File" for more information on the `agent.properties` file.

   > **Note:** Keep a copy of the `agent.properties` file so that you can reuse the configurations if you need to reinstall the agent at a later time. If you create multiple agents with different configurations, keep copies of the different `agent.properties` files for each configuration.

4. Execute the `wsmadmin installAgent` command.

See "installAgent" on page C-22 in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information about this command.

Redeploying the Web Service Client

Redeploy the Web service client for the changes to take effect.

Installing a BPEL or ESB Client Agent

The following is an overview of the steps to install a client agent for a BPEL or ESB process:

1. Add a client agent.
2. Define a policy for the client agent.
3. Install the client agent.
4. Add the `oracle.wsm.agent` shared library.
5. Deploy the client agent in the BPEL or ESB process.

The sections that follow describe each of these steps in detail.

To add a client agent

1. Use the Web Services Manager Control to create a client agent, and select the following values from the drop-down lists.
   a. **Component type** – Client Agent
   b. **Container type** – OC4J
2. Select **Register.** This generates a component ID. Make a note of this component ID because you will use this ID when you install the client agent.

To define a policy for the client agent

Configure the policy you want to associate with the agent. See Chapter 5, "Oracle Web Services Manager Policy Management," in *Oracle Web Services Manager Administrator's Guide* for more information on how to define a policy for an agent.

To install the client agent

1. Edit the following attributes in the *`ORACLE_HOME`*`/owsm/bin/agent.properties` file:
   - `agent.componentType` – OC4JClientInterceptor
   - `client.home` – Enter the path to the home directory for OC4J.
   - `agent.component.id` – Enter the component ID that was generated when the agent is created and registered using Web Services Manager Control.

   Note: Keep a copy of the `agent.properties` file so that you can reuse the configurations if you need to reinstall the agent at a later time. If you create multiple agents with different configurations, keep copies of the different `agent.properties` files for each configuration.

2. Execute the `wsmadmin installAgent` command.

A directory, similar to *CLIENT_HOME*\owsm\config\interceptors\component_ID is created that contains the client configuration. Make a note of this directory.

See "installAgent" on page C-22 in Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information about this command.

Add the oracle.wsm.agent shared library

Open the *OC4J_HOME*/j2ee/home/config/server.xml file and search for the following line:

```
<shared-library name="oracle.wsm.agent" version="10.1.3.1"
library-compatible="true">
```

If you do not see a shared library with the name oracle.wsm.agent, then you must complete a workaround to add the required library. The workaround is to install a temporary server agent, which adds the oracle.wsm.agent shared library to the server.xml file. Once the library has been added to the file, the temporary server agent is deleted.

1. Edit the following properties in the *OC4J_HOME*/owsm/bin/agent.properties file:
   - agent.componentType – OC4JServerInterceptor
   - agent.component.id – C0009999

     The value for the agent.component.id is an arbitrary identifier for the temporary server agent.
2. Verify that the Oracle Application Server is running.
3. Execute the following command to create a temporary server agent:

   ```
   ORACLE_HOME/owsm/bin/owsmadmin installAgent
   ```

   The command adds the shared library and creates a directory for the server agent.
4. Delete the temporary server agent by deleting the *OC4J_HOME*/owsm/config/interceptors/C0009999 directory.

Deploy the client agent in the BPEL or ESB process

1. Create the wsif-wsm-config.xml file using the oracle-webservices-client-10_0.xsd schema. The schema is located in the *OC4J_HOME*/j2ee/home/lib/oc4j-schemas.jar file.

   Refer to the *Advanced Web Services Developer's Guide* for more information.
2. Add the location of the client agent runtime to wsif-wsm-config.xml file.

   The following is an example of the wsif-wsm-config.xml file with the location of the client agent runtime in bold.

   ```
   <oracle-webservice-clients>
     <webservice-client>
       <service-qnamenamespaceURI=
        namespaceURI="http://oracle.com/esb/namespaces/DefaultSystem"
        localpart="CalcService_CA"/>
       <port-info>
        <wsdl-port namespaceURI="http://oracle.com/esb/namespaces/DefaultSystem"
          localpart="CalcService_CASoapHttpPort"/>
         <runtime enabled="owsm">
         <owsm init-home="C:\Oracle\product\owsm\config\interceptors\C0003003"/>
   ```

```
      </runtime>
      <operations/>
    </port-info>
  </webservice-client>
</oracle-webservice-clients>
```

Note: There is one `wsif-wsm-config.xml` file for your entire Oracle Application Server installation. If you want to install client agents into multiple BPEL or ESB processes, you must provide the `<service-name>` and `<wsdl-port>` tags for each process for which you want to install a client agent.

3. Save the `wsif-wsm-config.xml` file in the *`OC4J_HOME`*`/j2ee/home/config` directory.
4. Restart the Oracle Application Server.

The agent.properties File

Edit the *`ORACLE_HOME`*`/owsm/bin/agent.properties` file to customize the server or client agent. This section describes the properties in the `agent.properties` file.

Note: The property values are case-sensitive.

Table 6–1 agent.properties File

Property	Description
`agent.componentType`	Valid values are: `serveragent`, `clientagent`, `OC4JServerInterceptor`, or `OC4JClientInterceptor`
`agent.containerType`	Specify `OC4J` or `AXIS`.
`agent.containerVersion`	Specify `10.1.3.1` for all agents, except agents for AXIS Web services and AXIS clients. Specify `1.1` or `1.4` for AXIS agents.
`oc4j.home`	Path to a valid OC4J install directory, for example, `c:\ORACLE_HOME`. This property is required only for server agents installed for an Oracle Web service.
`oc4j.j2ee.home`	Path to the OC4J J2EE installation directory.
`external.oc4j.port`	Port on which OC4J listens for HTTP requests.
`external.oc4j.adminPort`	RMI port. The value for this property is automatically added after the installation. For example, the default value is `23791`. If you have installed the Oracle SOA Suite, this value is the OPMN port number. This property is required only for server agents installed for an Oracle Web service and client agents installed for a J2EE client.

Table 6–1 (Cont.) agent.properties File

Property	Description
external oc4j.adminID	OC4J Administrator user. For example, oc4jadmin. This property is required only for server agents installed in an Oracle Web service and client agents installed for a J2EE client.
client.home	Path to a valid client home directory. This is where the client agent will live. This property is required only for client agents.
webservice.application.input	Full path of the application EAR or WAR file. If you supply the full path to the EAR file, then you must specify the webservice.application.webapp.name property. This property is required only for server agents installed for an AXIS Web service and client agents installed for an AXIS client.
webservice.application.webapp.name	By default, this property is commented out. If you specify the full path for an EAR file, then uncomment this property and specify the name of the application. This property is required only for server agents installed for an AXIS Web service and client agents installed for an AXIS client.
webservice.application.contexturi	Context URI within the application server. This property is required only for server agents installed for an AXIS Web service and client agents installed for an AXIS client.
agent.installDir	Directory into which the agent is installed. This property is required only for TIBCO Business Works agents. This property is not supported in the Oracle WSM 10.1.3.1.0 release.
agent.traPath	Path to the TIBCO Business Works tra file. This property is required only for the TIBCO Business Works agents. This property is not supported in the Oracle WSM 10.1.3.1.0 release.
agent.component.id	The value for this property is generated when you register the agent in the Oracle WSM Policy Manager.
agent.policymanager.enabled	If you specify true, then the agent gets policies from the Oracle WSM Policy Manager in online mode. If you set this property to true, then you must specify values for the agent.policymanagerURL, policySet.poll.enabled, and policySet.poll.frequency properties. If false, then the agent retrieves policies from a file (offline mode).
agent.policymanagerURL	Location of the Oracle WSM Policy Manager URL. Set this property if agent.policymanager.enabled is set to true.

Table 6–1 (Cont.) agent.properties File

Property	Description
`policySet.poll.enabled`	Specify `on` or `off` to enable or disable polling of the Oracle WSM Policy Manager. Set this property if `agent.policymanager.enabled` is set to `true.`
`policySet.poll.frequency`	Polling frequency in seconds. Set this property if `agent.policymanager.enabled` is set to `true.`
`agent.policySet.file`	Name of the default policy set. This property is required if Oracle WSM is run in offline mode, that is, when `agent.policymanager.enabled` is set to `false.`
`agent.policySet.propertiesFile`	Name of the property file packaged with the Web application which specifies the location of the policy set file.

A

Troubleshooting

This appendix provides information for troubleshooting your Oracle Web Services Manager (Oracle WSM) deployment.

Failed to Retrieve Policy Set Error Message

Problem

You install Oracle WSM and start Oracle Application Server. The following error appears in the *ORACLE_HOME*/j2ee/*instance*/log/gateway.log file:

```
Failed to retrieve policy set from policy manager with url
@ http://host_name:port_number/policymanager/services/RegistrationService:
com.cfluent.policymanager.sdk.base.exception.ServerException: java.lang.Exception:
Invalid component ID - C0003001"
```

(In the path to the gateway.log file, the variable *instance* is the name of the OC4J instance into which Oracle WSM is installed. If you install the standalone version of Oracle WSM, the default value of the instance is *home*.)

Solution

By default, at the time you install Oracle WSM, the component ID for the gateway is set to C0003001 in the gateway-config-installer.properties file. Therefore, when you start Oracle Application Server, the Oracle WSM tries to retrieve the policies for the C0003001 component. Because the gateway has not yet been created, the policies cannot be retrieved and an error is logged.

Log in to Web Services Manager Control, and add a gateway to Oracle WSM. Make the gateway the first policy enforcement point you create. This component will be assigned the component ID C0003001. Once the component has been added, the error message will no longer appear.

Error When Logging In to Web Services Manager Control

Problem

You log in to the Web Services Manager Control and get the following error:

```
The following exception occurred when processing the JSP:
org.xml.sax.SAXException: Bad envelope tag: HTML
Use your browsers "Back" button if you would like to try again.
```

Solution

This is a known bug that occurs after you deploy the Oracle WSM Policy Manager. Restart the Oracle Process Manager and Notification Server (OPMN) process (`opmnctl restartproc`) on the OC4J instance where the Oracle WSM components are installed. Then retry logging in to Web Services Manager Control.

B

Oracle Web Services Manager Configuration Files

This appendix is a reference for the Oracle WSM configuration files referred to in this guide.

Oracle Web Services Manager Configuration Files and Locations

Once you have installed Oracle Web Services Manager (Oracle WSM), you edit configuration files to customize Oracle WSM components. Table B–1 lists the names and locations of these files:

Table B–1 Name and Location of Oracle Web Services Manager Configuration Files

Name	Location
`agent.properties`	*ORACLE_HOME*/owsm/bin
`clientagent-config-installer.properties`	*ORACLE_HOME*/owsm/config/clientagent/
`collector-config-installer.properties`	*ORACLE_HOME*/owsm/config/coreman
`coresv.properties`	*ORACLE_HOME*/owsm/config/bin
`gateway-config-installer.properties`	*ORACLE_HOME*/owsm/config/gateway/
`install.properties`	*ORACLE_HOME*/owsm/bin/
`manageUserGroups.properties`	*ORACLE_HOME*/owsm/bin/
`monitor-config-installer.properties`	*ORACLE_HOME*/owsm/config/coreman
`monitorui-config-installer.properties`	*ORACLE_HOME*/owsm/config/ccore
`oc4j.xml`	*ORACLE_HOME*/owsm/scripts/
`opmn.xml`	*ORACLE_HOME*/opmn/conf/
`policymanager-config-installer.properties`	*ORACLE_HOME*/owsm/config/policymanager
`policyui-config-installer.properties`	*ORACLE_HOME*/owsm/config/ccore
`serveragent-config-installer.properties`	*ORACLE_HOME*/owsm/config/serveragent/

Table B–1 (Cont.) Name and Location of Oracle Web Services Manager Configuration

Name	Location
`ui-config-installer.properties`	*`ORACLE_HOME`*`/owsm/config/ccore`

C

Oracle Web Services Manager WSMADMIN Commands

The chapter provides an overview of the capabilities of the Oracle Web Services Manager (Oracle WSM) `wsmadmin` commands.

- Registering the required enforcement components.
- Installing additional applications, and enforcement components.

Overview of the WSMADMIN Commands

Once you have installed Oracle WSM, use the information in this chapter to help use the Oracle WSM Configuration Assistant to configure and customize your installation to your topology requirements.

Location of the WSMADMIN commands

Execute the wsmadmin command from the following directory:

```
ORACLE_HOME\owsm\bin
```

Using the WSMADMIN Command on UNIX

When using the Oracle WSM WSMADMIN commands on UNIX operating systems, the command must be specified as `wsmadmin.sh`. The Oracle WSM documentation uses the syntax for Microsoft Windows operating systems (that is, `wsmadmin`) for all WSMADMIN commands. If you are using the commands on a UNIX system, replace `wsmadmin` with `wsmadmin.sh`.

Using the wsmadmin start Command to Start the Oracle WSM Server

For the standalone Oracle WSM installation, the Oracle WSM server should be started using the `wsmadmin start` command. See "start" on page C-25.

> **Note:** Oracle WSM does not work if the bundled application server is started independently.

If you have installed Oracle WSM as part of the Oracle Application Server 10g release 3 (10.1.3.1.0) release, refer to *Oracle Application Server Administrator's Guide* for more information on starting the server.

Syntax of the WSMADMIN Commands

The following is the syntax for the `wsmadmin` command:

```
wsmadmin operation parameters
```

The following are the valid values for *operation*:

- applyTopology
- buildApps
- configApps
- copyDBConfig
- dataGenerator
- dataload
- dataloadConfigure
- deploy
- deployApps
- encodePasswords
- exportDBData
- exportPolicySet
- exportTransferableObjects
- help
- importTransferableObjects
- initialize
- inspectTopology
- install
- installAgent
- installOC4J
- installOLite
- manageUsergroups
- md5encode
- migrate
- purgePolicies
- start
- startOC4J
- startOLite
- stop
- stopOC4J
- stopOLite
- undeploy
- uninstall

- uninstallOC4J
- uninstallOLite
- upgrade

The commands are case-sensitive. Each operation is described in the following pages.

Passwords for WSMADMIN Commands

In order to execute many of the WSMADMIN commands, you must supply a password for the application server, the database server, or both. When you execute the command, you will be prompted to provide the required password. When you provide the password in this manner, the password is encoded and secure.

There may be times when you need to automate a process and provide the password without direct user input. In such situations, you can provide passwords on the command line using the following syntax:

- `-Doc4jAdminPassword=`*`password`* – Use this parameter to supply the password for Oracle Application Server.
- `-DDBPassword=`*`password`* – Use this parameter to supply the password for Oracle Database Server.

> **Caution:** When you provide the password on the command line, it is not secure and can be viewed by any Oracle WSM user.

You can also supply the password to a script using input redirection operators.

applyTopology

This operation creates a clone of Oracle WSM.

Syntax

```
wsmadmin applyTopology
```

Usage Notes

- This command uses the `topology.properties` file that is generated with the `inspectTopology` command to create a clone of Oracle WSM. If you clone the Oracle Application Server, the topology folder from the source machine is copied over to the correct location on the destination machine. Otherwise, the folder will have to be manually copied to *`ORACLE_HOME`*`\owsm\config` on the destination machine.
- Before you use the `wsmadmin applyTopology` command, you must edit the `topology.properties` file with the topology information for the destination machine. See "The topology.properties File" for information about the contents of this file and how to update this information.
- If the `database.update` property is set to `true`, then the `applyTopology` command deletes any Oracle WSM data in the Oracle Database on the destination machine. Then it copies the tables from the *`ORACLE_HOME`*`/owsm/config/topology/export` and *`ORACLE_HOME`*`/owsm/config/topology/template` directories on the source machine to the *`ORACLE_HOME`*`/owsm/config/topology/import` directory on the destination machine. It applies the necessary changes to the database tables.

Finally, it imports the database tables into the destination installation and redeploys the Oracle WSM components.

- If your destination installation is distributed across multiple systems, then the first time you execute this command, the `database.update` property in the `topology.properties` file should be set to `true` to import the database tables. When you clone Oracle WSM components on subsequent machines, the `database.update` property should be set to `false`.

The topology.properties File

The file containing the description of the source installation topology is: *`ORACLE_HOME`*`/owsm/config/topology/topology.properties`. The following table is a partial list of properties in this file. Verify the values for these properties and change them, as necessary, so that they are appropriate for the destination machine. For example, if the source and destination installations are on different machines, at a minimum, you will have to change the host names and ports for the components.

If the database on the destination installation is different than the source installation, the edit the database properties in the file. Enter the database password for the `database.password` property in clear text, then use the `wsmadmin encodePassword` command to encrypt the password.

Table C–1 topology.properties Properties

Property Name	Description

`control.instance`
`gateway.instance`
`monitor.instance`
`policymanager.instance`
	Specifies the OC4J instance in which the Oracle WSM components are redeployed on the destination installation.
	If this property is not set, then the particular component is not redeployed on the destination machine.

`database.update`	Specifies whether the Oracle Database is cloned. The default value is false.

`database.url`
`database.driver`
`database.user`
`database.password`
	Properties that specify the Oracle Database used by Oracle WSM.
	Use the `wsmadmin encodePasswords` command to encrypt the database password.

`monitor.rmi.host`
`monitor.rmi.port`
`monitor.soap.protocol`
`monitor.soap.host`
`monitor.soap.port`
	Properties that specify the Oracle WSM Monitor RMI and SOAP endpoint.

`monitor.notification.smtp.host`
`monitor.notification.smtp.port`
`monitor.notification.smtp.user`
`monitor.notification.smtp.password`
`monitor.notification.smtp.from`
	Properties that specify the SMTP server and credentials for Oracle WSM Monitor alerts.

Table C–1 (Cont.) topology.properties Properties

Property Name	Description

`monitor.notification.snmp.host`
`monitor.notification.snmp.port`
`monitor.notification.snmp.authProtocol`
`monitor.notification.snmp.user`
`monitor.notification.snmp.password`
	Properties that specify the SNMP server and credentials for Oracle WSM Monitor alerts.

`control.protocol`
`control.host`
`control.port`
	Properties that specify the Web Services Manager Control endpoint.

`control.auth.provider`
`control.auth.ldap.host`
`control.auth.ldap.port`
`control.auth.ldap.openPort`
`control.auth.ldap.sslPort`
	Properties that specify the LDAP directory against which Web Services Manager Control authenticates users.

`control.help.location`	Property that specifies the location of external online help files.
`control.contact.url`	Specifies the URL for contacting your technical support organization.
`gateway.component.id` `gateway.policymanagerURL`	Properties that specify the Oracle WSM Gateway endpoint.

`policymanager.protocol`
`policymanager.host`
`policymanager.port`
	Properties that specify the Oracle WSM Policy Manager endpoint.

In addition, you will find additional properties from the component configuration files are added to the `topology.properties` file, depending on the particular components, policy enforcement points, and policies in your destination installation. For example, you may find the following categories of properties:

- Oracle WSM Gateway Context URL Property
- Web Service WSDL Property
- Policy Enforcement Point Properties
- Component Configuration Properties
- Policy Pipeline Step Properties

> **Note:** In the syntax statements, bold text indicates the actual text you see in the entries.

Oracle WSM Gateway Context URL Property The following property defines the context URL for each Oracle WSM Gateway:

component-component_ID.**contextUrl=***url*

where:

- *component* – Name of the component
- *component_ID* – ID of the component
- *url* – URL to the Oracle WSM Gateway

For example,

```
gateway-C0003001.contextUrl=http://10.178.95.103:80/gateway
```

Web Service WSDL Property Each Web Service WSDL is defined with the following property:

component_name–*component_ID*.**service**–*service_ID*.**wsdl_url**=*url*

where:

- *component_name* – Name of the component, for example, gateway
- *component_ID* – ID of the component
- *service_ID* – ID of the Web service
- *url* – URL for the Web service endpoint

For example:

```
gateway-C0003001.service-SID0003001.wsdl_
url=http://10.178.95.103:80/ccore/TimeService.wsdl
gateway-C0003001.service-SID0003002.wsdl_url=http://10.178.95.103:80/6149927_
Client-HelloWorld_WS-context-root/MyWebService1Soap12HttpPort?WSDL
gateway-C0003001.service-SID0003003.wsdl_
url=http://10.178.95.103:80/CDIServicesWS/CDIServicesWSSoapHttpPort?WSDL
```

Policy Enforcement Point Properties Each Oracle WSM component has a set of default properties. These properties are defined with the following syntax:

default.*property*=*value*

- *property* – property name
- *value* – value assigned to the property

where:

```
default.cfluent.messagelog.enabled=true
default.cfluent.messagelog.type=db
default.cfluent.messagelog.file.maxFileSize=5000000
default.cfluent.messagelog.db.url=${database.url}
default.cfluent.messagelog.db.driver=${database.driver}
default.cfluent.messagelog.db.userid=${database.userid}
default.cfluent.messagelog.db.password=${database.password}
...
```

These default settings cannot be changed once the Oracle WSM installation is cloned. Therefore, if you want to change the defaults, you must make any changes to these properties in the `topology.properties` file.

Component Configuration Properties There is a set of properties that specify the configuration for each policy enforcement point with the following syntax:

pep–*component_ID*–**v***version*–*property*=*value*

where:

- *component_ID* – Component ID
- *version* – Version of the component
- *property* – Name of the component property
- *value* – Value assigned to the property

```
pep-C0003003-v26-cfluent.messagelog.enabled=true
pep-C0003003-v26-cfluent.messagelog.type=db
pep-C0003003-v26-cfluent.messagelog.file.logDirectory=
pep-C0003003-v26-cfluent.messagelog.file.maxFileSize=5000000
pep-C0003003-v26-cfluent.messagelog.db.url=${database.url}
pep-C0003003-v26-cfluent.messagelog.db.driver=${database.driver}
pep-C0003003-v26-cfluent.messagelog.db.userid=${database.userid}
...
pep-C0003002-v58-cfluent.messagelog.enabled=true
```

Policy Pipeline Step Properties A comment identifies the type of policy step. This is followed by a series of properties describing the configuration of the policy step:

pep–component_ID–*policy_version*–*service_name* (*service_version*)–*pipeline_type*.**step***step_number*.*property*= *value*

where:

- *component_ID* – ID of the pipeline
- *policy_version* – Version number of the policy
- *service_name* – Name of the Web service
- *service_version* – Version of the Web service
- *pipeline_type* – Specifies the pipeline to which the policy step belongs. Valid values are Request or Response.
- *step_number* – ID of the step
- *property* – Name of the property
- *value* – Value assigned to the property

```
#pipeline3047.step0.implementation=com.cfluent.policysteps.common.LoggerStep
pep-C0003002-v3-OpenTradeService(1.0)-Response.step0.Enabled=true
pep-C0003002-v3-OpenTradeService(1.0)-Response.step0.LogLevel=envelope
#pipeline3049.step0.implementation=com.cfluent.policysteps.common.LoggerStep
pep-C0003002-v3-OpenTradeService(1.0)-Request.step0.Enabled=true
pep-C0003002-v3-OpenTradeService(1.0)-Request.step0.LogLevel=envelope
#pipeline3049.step1.implementation=com.cfluent.policysteps.security.wssecurity.Sig
nStep
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.Enabled=false
#pipeline3049.step1.implementation=com.cfluent.policysteps.security.wssecurity.Sig
nStep
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.Enabled=false
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.KeyStoreFile=/my/keystore/loca
tion
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.KeyStoreType=jks
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.KeyStorePassword=????BUUpfArOY
7Dqr32ZiBi0f9A=
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.SignerKeyAlias=alias
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.SignerKeyPassword=????BSp5xQ0l
eeFHs3CvuBTlcng/EqyYAv72kA==
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.SignatureAlgorithm=RSA-SHA1
```

```
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.SignedContent=BODY
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.XpathExpression=
pep-C0003002-v3-OpenTradeService(1.0)-Request.step1.NameSpaces=
vt
```

Related Commands

inspectTopology

buildApps

This operation uses the JAR and JSP source files to build the J2EE application WAR or EAR files that are deployed.

Syntax

```
wsmadmin buildApps
```

Usage

The values for the installed applications are saved to the `install.properties` file during the Oracle WSM installation. After you have performed the `buildApps` operation, you need to perform the `deployApps` operation. There may be situations when you need to individually build and deploy your applications. For example, you may need to perform the following manual tasks between the `buildApps` and `deployApps` operations:

- Version the Oracle WSM application archives before they are deployed
- Prepare the application archives to be deployed manually

When you have finished building applications with the `buildApps` operation, use `deployApps` to deploy the application archives using the standard settings.

configApps

This operation generates new application configuration files based on the values in the `install.properties` file.

Syntax

```
wsmadmin configApps
```

Usage

Use this operation before running `buildApps` or `deploy`. Performing this operation does not affect JSSO-enabled applications. You may use this operation to change a database password and then redeploy the application.

This operation reads the following properties from the `install.properties` file:

Table C–2 install.properties Used During configApps

Attribute Name	Description
`install.appserver.type`	Select the Application Server type as `oracle-as`.
`install.http.host`	Specify the HTTP host name.
`install.http.port`	Specify the HTTP port.

Table C–2 (Cont.) install.properties Used During configApps

Attribute Name	Description
`install.oc4j.instance`	Specifies the name of the container to deploy the application.
`install.sso.support`	Specify the value as `true`, only if you enable single sign-on for the Oracle WSM Control.
`install.proxy.host`	Name of the proxy server.
`install.proxy.port`	Port number of the proxy server.
`install.noproxy.hosts`	List of hosts where you may not connect using the proxy server. You can create a list separated by a vertical bar (|), and use the asterisk (*), for wildcard matching.
`install.db.type`	Set the value to `oracle`, or `olite`.
`install.db.driver.type`	Specify the driver type, either `thick` or `thin`.
`install.db.host`	Specify the machine name where the database is installed or the IP address.
`install.db.port`	Specify the port number for the database.
`install.db.userid`	Specify the database user ID.
`install.db.password`	Specify the password for the database.
`sqlserver.jdbc.jars.path`	Specify the location of the SQL Server JDBC JAR files.

Performing this operation overwrites the existing files and creates the following files:

ORACLE_HOME`\owsm\bin\agent.properties`

ORACLE_HOME`\owsm\bin\coresv.properties`

ORACLE_HOME`\owsm\config\ccore\logging.xml`

ORACLE_HOME`\owsm\config\ccore\logging.xml`

ORACLE_HOME`\owsm\config\ccore\ui-config-installer.properties`

ORACLE_HOME
`\owsm\config\clientagent\clientagent-config-installer.properties`

ORACLE_HOME`\owsm\config\clientagent\logging.xml`

ORACLE_HOME`\owsm\config\coreman\logging.xml`

ORACLE_HOME
`\owsm\config\coreman\monitor-config-installer.properties`

ORACLE_HOME
`\owsm\config\gateway\gateway-config-installer.properties`

ORACLE_HOME`\owsm\config\gateway\logging.xml`

This operation creates all the files necessary to configure all the applications, including the subset of files created when you use the `copyDBConfig` operation.

You need to redeploy the applications to propagate the changes.

copyDBConfig

This operation copies the database configuration parameters into the configuration directory property files.

Syntax

```
wsmadmin copyDBConfig
```

Syntax

Use this operation before `buildApps` or `deploy`. This operation reads the following values from the `coresv.properties` file:

Table C–3 coresv.properties Used During copyDBConfig

Attribute Name	Description
`dataload.messagelog.db.url`	Location of the database URL.
`dataload.messagelog.db.driver`	Type of database driver, `thick` or `thin`.
`dataload.messagelog.db.userid`	User ID used to login to the database.
`dataload.messagelog.db.password`	Encrypted password used to log in to the database.

After reading the values, the operation modifies the following `coresv.properties` files:

ORACLE_HOME`\owsm\bin\agent.properties`

ORACLE_HOME`\owsm\bin\coresv.properties`

ORACLE_HOME`\owsm\config\ccore\ui-config-installer.properties`

ORACLE_HOME
`\owsm\config\clientagent\clientagent-config-installer.properties`

ORACLE_HOME
`\owsm\config\coreman\monitor-config-installer.properties`

ORACLE_HOME
`\owsm\config\gateway\gateway-config-installer.properties`

You need to redeploy the applications to propagate the changes.

dataGenerator

This operation is a test tool that generates the data you can view on the Oracle WSM Monitor. Use `wsmadmin dataGenerator` to view the following types of data:

- ping data
- flow data
- invocation data.

Syntax

```
wsmadmin dataGenerator
```

Usage

Use `wsmadmin dataGenerator` to view the following types of data:

- ping data
- flow data
- invocation data.

The following properties are exclusively for the Oracle WSM Monitor. This operation reads the following values from the `coresv.properties` file:

Table C–4 coresv.properties Used During dataGenerator

Attribute Name	Description
`dataload.monitor.rmi.host`	Host name where the Oracle WSM Monitor is installed.
`dataload.monitor.rmi.port`	Port number where the Oracle WSM Monitor is installed.

dataload

This operation loads SQL scripts generated by the `dataloadConfigure` operation into the primary Oracle WSM Database.

Syntax

```
wsmadmin dataload
```

Usage

The `dataload` operation checks whether the data already exists in the database before loading the data into the Oracle WSM Database. If data exists, then the operation exits.

You must perform the `dataloadConfigure` operation before performing `dataload`. The two operations `dataload` and `dataloadConfigure` provide the flexibility to customize the data before loading it into the database.

The `initialize` operation combines the `dataload` and `dataloadConfigure` operations.

The `dataload` operation reads the following attributes from the `coresv.properties` file:

Table C–5 coresv.properties Used During dataload

Attribute name	Description
`dataload.messagelog.db.url`	Location of the database URL.
`dataload.messagelog.db.driver`	Type of database driver, `thick` or `thin`.
`dataload.messagelog.db.userid`	User ID used to log in to the database.
`dataload.messagelog.db.password`	Encrypted password to log in to the database.

You will be prompted to provide a password to Oracle WSM Database. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

dataloadConfigure

This operation generates SQL scripts that are used to create the database schema and load seed data.

Syntax

```
wsmadmin dataloadConfigure
```

Usage

The database files are created in *ORACLE_HOME*\owsm\db. The dataloadConfigure operation reads the following attributes from the coresv.properties file:

Table C–6 coresv.properties Used During dataloadConfigure

Attribute name	Description
dataload.messagelog.db.url	Location of the database URL.
dataload.messagelog.db.driver	Type of database driver, thick or thin.
dataload.messagelog.db.userid	User ID used to log in to the database.
dataload.messagelog.db.password	Encrypted password to log in to the database.

Perform the dataload operation after you have performed dataloadConfigure. For more information, see "dataload" on page C-11.

The initialize operation combines the dataload and dataloadConfigure operations.

deploy

This operation builds, deploys, and binds components to an OC4J application engine.

Syntax

```
wsmadmin deploy component
```

Parameters

- *component* – Name of the component being deployed. The valid values are:
  - all – Deploys all components
  - control – Deploys the Web Services Manager Control
  - gateway – Deploys the Oracle WSM Gateway
  - monitor – Deploys the Oracle WSM Monitor
  - policymanager – Deploys the Oracle WSM Policy Manager

Usage

Use this operation to build, deploy, bind controls, gateways, monitors, or policy manager instances. You can perform these tasks on all the components at the same time. Perform this operation to deploy applications without customizing them before deployment. To customize applications before deploying, see "buildApps" on page C-8.

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

If you are using Oracle Process Manager and Notification Server (OPMN) to manage Oracle WSM and you deploy Oracle WSM Policy Manager, then you must restart the OPMN process on the OC4J instance where the Oracle WSM components are installed. If you do not restart OPMN, you will get an error when you try to log in to Web Services Manager Control.

deployApps

This operation deploys components to an OC4J application engine.

Syntax

```
wsmadmin deployApps component
```

Parameters

- *component* – Name of the component being deployed. The valid values are as follows:
  - `all` – Deploys all components
  - `control` – Deploys the Web Services Manager Control
  - `gateway` – Deploys the Oracle WSM Gateway
  - `monitor` – Deploys the Oracle WSM Monitor
  - `policymanager` – Deploys the Oracle WSM Policy Manager

Usage

Use this operation to deploy instances of Web Services Manager Control, Oracle WSM Gateway, Oracle WSM Monitor, or Oracle WSM Policy Manager. You can deploy all of the components at once using the `all` parameter or you can deploy the individual components. Use this operation after using the `buildApps` operation. For more information see "buildApps" on page C-8. The `deployApps` operation reads the following attributes from the `coresv.properties` file:

Table C–7 coresv.properties Used During deployApps

Attribute Name	Description
`oc4j.j2ee.home`	Location of the OC4J container.
`external.oc4j.home`	Location of the external OC4J installation.
`external.oc4j.host`	Host name for the external OC4J installation.
`external.oc4j.port`	Port number for the external OC4J installation.
`external.oc4j.adminPort`	Port number for the administrator of the external OC4J installation.
`external.oc4j.adminID`	Administrator's user name for the external OC4J installation.
`external.oc4j.instance`	Location of the OC4J instance.

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

If you are using Oracle Process Manager and Notification Server (OPMN) to manage Oracle WSM and you deploy Oracle WSM Policy Manager, then you must restart the OPMN process on the OC4J instance where the Oracle WSM components are installed. If you do not restart OPMN, you will get an error when you try to log in to Web Services Manager Control.

encodePasswords

This operation encodes passwords in a property file.

Syntax

```
wsmadmin encodePasswords propertyFile properties
```

Parameters

- *propertyFile* – Name of the specified file
- *properties* – List of passwords that will be encoded

Usage

You can separate multiple passwords with a colon (:). However, spaces are not permitted.

exportDBData

This operation exports data from the Oracle WSM Database to a file.

Syntax

```
wsmadmin exportDBData
```

Usage

The exported tables are stored as XML files. You may perform this operation for the following reasons:

- To back up your Oracle WSM data.
- To store your data before moving your data to different location.

> **Note:** Check that the PATH environment variable is set to the location where SQL*Plus is installed.

The `exportDBData` operation reads the following attributes from the `coresv.properties` file:

Table C–8 coresv.properties Used During exportDBData

Attribute Name	Description
`db.export.dir`	Default location where you want to store your Oracle WSM Database file is `c:\temp`. Change this value to the location of your choice.

You will be prompted to provide a password to Oracle WSM Database. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

When you execute the command, the following message is displayed:

```
"Warning!!! The data in the input directory will be deleted. Are
you sure you want to continue? Y-Yes N-No:"
```

Select `Y` to proceed with the data export.

exportPolicySet

This operation exports policies associated with a component.

Syntax

```
wsmadmin exportPolicySet component_ID
```

- *component_ID* – the component ID of the Oracle Web Services Gateway or Agent whose policies are being exported.

Usage

- The command packages the policies with the policy enforcement point so that the Oracle WSM Gateway or Agent does not have to communicate with Oracle WSM Policy Manager to get the policies.
- The Oracle WSM instance must be up and running when this command is executed.
- Before you use the exportPolicy command, you must add several properties to the *ORACLE_HOME*/owsm/bin/coresv.properties file.

Table C–9 coresv.properties Used with exportPolicySet

Property	Description
pm.server.httpScheme	Transport on which Policy Manager is running. The valid values are http and https.
pm.server.httpHost	Name of the host on which Policy Manager is running
pm.server.httpPort	Port on which Policy Manager listens for requests.

- The policy is exported to the directory defined by the db.export.dir property which is set in the *ORACLE_HOME*/owsm/bin/coresv.properties file. The name of the file is generated dynamically as PolicySet–*component_ID*–*timestamp*.xml. The variable, *component_ID* is the component ID specified in the wsmadmin exportPolicySet command. The timestamp of the export is appended to the file name to further identify the exported policy.

exportTransferableObjects

This operation exports selected Oracle WSM objects.

Syntax

```
wsmadmin exportTransferableObjects
```

Usage Notes

- The wsmadmin exportTransferableObjects command uses the *ORACLE_HOME*\scripts\LMTInstructions.xml file and exports the Oracle WSM objects specified in the file.
- You must specify the ID of the components, services, and custom step templates that you want to export in the LMTInstructions.xml file. In addition, you must specify the version of the step template being exported. See Chapter 10, "Oracle WSM Cloning and Horizontal Migation," in *Oracle Web Services Manager Administrator's Guide* for an example of the input file.
- Set the db.export.dir property in *ORACLE_HOME*\owsm\bin\coresv.properties file to the directory where you want the components exported.

Related Commands

importTransferableObjects

help

This operation displays the list of `wsmadmin` operations.

Syntax

```
wsmadmin help operation
```

Parameter

`operation` – Any valid `wsmadmin` operation described in this appendix. See "Syntax of the WSMADMIN Commands" on page C-2 for a complete list of `wsmadmin` operations.

Usage

If you execute the command `wsmadmin help,` you get a list of the valid operations you can perform with the command. To get help for a specific operation, including any arguments that the operation accepts, specify the name of the operation with the `operation` variable.

importDBData

This operation imports data from a file into the Oracle WSM Database.

Syntax

```
wsmadmin importDBData
```

Usage

Confirm that the following requirements have been met:

- Ensure that SQL*Plus is installed before performing this operation.
- Verify that the PATH environment variable is set to the location where SQL*Plus is installed.

The `importDBData` operation reads the following attributes from the `coresv.properties` file:

Table C–10 coresv.properties Used During importDBData

Attribute Name	Description
`db.import.dir`	Default location from where you want to transfer data to your Oracle WSM Database is `c:\temp`. You may change this value to another location.

You will be prompted to provide a password to Oracle WSM Database. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

importTransferableObjects

Imports Oracle WSM objects.

Syntax

```
wsmadmin importTransferableObjects
```

Usage

- The command uses the information in the `LMTInstructions.xml` file and imports the specified components, services, policies, and custom templates with the attributes specified in the file.
- Any imported Oracle WSM components, services, and step templates are assigned the next available ID on the destination machine.

LMTInstructions.xml File

The `LMTInstructions.xml` file must be in the *`ORACLE_HOME`*`/scripts` directory on the destination machine.

> **Note:** See Chapter 10, "Oracle WSM Cloning and Horizontal Migation," in *Oracle Web Services Manager Administrator's Guide* for an example of the input file.

Component Attributes You must specify attributes for each component being imported. The following is an example of a `< component>` tag and its attributes.

```
<component id="C0003029"
     import-name="ImportedGateway16Jan"
     url="http://jdoe-pc.abc.oracle.com:7780/gateway"
     monitor-rmi-host="localhost"
     monitor-rmi-port="3118"
     monitor-soap="http://localhost:7780/coreman/services/CoremanClient"
     monitor-type="rmi"
     database-url="jdbc:oracle:thin:@//10.177.34.170:1521/orcl.abc.oracle.com"
     database-driver="oracle.jdbc.driver.OracleDriver"
     database-user-id="ORAWSM"
     database-user-password="????BXbXX/cf08QHKYYAb6v4o/bb2r37tTw4Tw=="
     create-new="false"
     type="Gateway"
     mapped-to-component-id="C0003029">
...
</component>
```

Table C–11 describes the attributes that must be specified.

Table C–11 <component> Tag Attributes

Attribute Name	Description
`id`	ID of the exported component.
`import-name`	Name that is assigned to the imported component. Verify that this name will be unique on the destination machine. If another component with the same name already exists on the destination machine, a *-1* will be appended to the value specified by `import-name`.
`url`	URL of the component being imported.
`monitor-rmi-host`	RMI host name of the destination machine on which Oracle Web Services Monitor is installed. This attribute is required when `monitor-type` is set to `RMI`.

Table C–11 (Cont.) <component> Tag Attributes

Attribute Name	Description
`monitor-rmi-port`	RMI port on the destination machine on which Oracle Web Services Monitor listens for requests. This attribute is required when `monitor-type` is set to `RMI`.
`monitor-soap`	SOAP endpoint service URL on the destination machine where the Oracle WSM Monitor measurements are processed. By default, this is set to `http://localhost/coreman/services/CoremanMeasurementClient`. This attribute is required when `monitor-type` is set to `SOAP`.
`monitor-type`	Specifies method of communication with the Oracle Web Services Monitor. The valid values are `RMI` and `SOAP`. (Thesevalues are case-sensitive.) If monitor-type is `RMI`, the the `monitor-rmi-host` and `monitor-rmi-port` attributes are used. If `monitor-type` is `SOAP`, then the `monitor-soap` attribute is used.
`database-url`	URL of the database on the destination machine into which the data is imported. This URL is specified in the *ORACLE_HOME*\owsm\bin\coresv.properties file on the destination machine.
`database-driver`	Driver of the database on the destination machine into which the data is imported. The driver is specified in the *ORACLE_HOME*\owsm\bin\coresv.properties file on the destination machine.
`database-user-id`	Database user name for the database on the destination machine into which the data is imported. The database user name is specified in the *ORACLE_HOME*\owsm\bin\coresv.properties file on the destination machine.
`database-user-password`	Database password for the database on the destination machine into which the data is imported. The database password is specified in the *ORACLE_HOME*\owsm\bin\coresv.properties file on the destination machine
`create-new`	If set to `true,` then a new component is created on the destination machine and any services that were exported for this component are imported. If set to `false,` then a component is not created on the destination machine, and only the services associated with the component are imported to the component specified with the `mapped-to-component-id` attribute.
`type`	Specifies the component type being imported. Valid values are: Gateway, ClientAgent, and ServerAgent.
`mapped-to-component-id`	If `create-new` is set to `false,` then this attribute must be specified with the component ID of the component on the destination machine to which the services will be mapped.

Service Attributes You must specify attributes for each Web service being imported. The `<service>` tag is a child tag of the `<component>` tag. A `<component>` tag may have multiple `<service>` tags. The following is an example of the `<service>` tag and attributes.

```
<component id="C0003029"
...
     <service id="SID0003268"
              import-name="TimeService16Jan1"
```

```
               version="1.0"
               wsdl="http://jdoe-pc.abc.oracle.com:7780/ccore/TimeService.wsdl"
               url="http://jdoe-pc.abc.oracle.com:7780/ccore/TimeService.jsp">
          <pipeline-property pipeline-name="Service" step="0"
prop-name="ReplyTimeout" prop-value="300000"/>
          <pipeline-property pipeline-name="Request" step="0" prop-name="LDAPHost"
prop-value="localhost"/>
    </service>
...
</component>
```

Table C–12 describes the attributes that must be specified for each service being exported.

Table C–12 <service> Tag Attributes

Attribute Name	Description
`id`	ID of the exported service.
`import-name`	Name that is assigned to the imported service. Verify that this name will be unique on the destination machine. If another service with the same name already exists on the destination machine, a *-1* will be appended to the value specified by `import-name`.
`version`	Version number of the exported service that will be imported.
`wsdl`	URL of the service WSDL to be imported.

Property Values When you import a service to the destination machine, you may want to change some of the properties. For example, the name of the LDAP Host on the destination installation may differ from the source installation, and, therefore, you need to change the value assigned to the LDAPHost property. Or you may want to change the time out value. When you import the component, you can specify the property and the new value using the `<pipeline-property>` tag. The `<pipeline-property>` tag is a child of the `<service>` tag. Table C–13 describes the pipeline-property attributes.

Table C–13 <pipeline-property> Tag Attributes

Attribute Name	Description
`pipeline-name`	Name of the pipeline that is to be modified.
`step`	Name of the step that is to be modified.
`prop-name`	Name of the property that is to be modified.
`prop-value`	Value to be assigned to the prop-name property on the destination machine.

Custom Template Attributes The `<step-template>` tag is a child of the `<component>` tag. When you import a step template into the destination Oracle Web Services Manager instance, you must assign an ID, name, and version number to the step template. The step template ID and name must be unique in the destination Oracle Web Services Manager instance. The following is an example of the <step-template> tag.

```
<component id="C0003030"
...
     <step-template id="891B611A-0F24-85DD-3C27-8B56B5F7G4F21248"
          version="1.0"
          import-id="891B611A-0F24-85DD-3C27-8B56B5F7G4F21249"
          import-name="Custom JMS Header step - XXXXXX"
          import-version="1.0"
     </step-template>
</component>
```

Table C–14 describes the attributes that must be specified for the `<step-template>` tag.

Table C–14 <step-template> Tag Attributes

Attribute Name	Description
`id`	ID of the exported step template
`version`	Version number of the exported step template.
`import-id`	ID of the imported step template. This number must be unique in the destination Oracle Web Services Manager instance.
`import-name`	Name of the imported step template. This name must be unique in the destination Oracle Web Services Manager instance.
`import-version`	Version number of the imported step template. Assign a version number to the step template that is being imported.

Related Commands

exportTransferableObjects

inspectTopology

This operation inspects an Oracle WSM installation.

Syntax

```
wsmadmin inspectTopology -D inspect.level=inspect_level_value
```

Parameters

- *`inspect_level_value`* – *0* (the default) inspects only the latest version of a policy, and *1* inspects all versions of a policy. The result of this inspection is recorded in the `topology.properties` file.

Usage Notes

- This command generates a file that describes the topology of the Oracle WSM installation, and expports the Oracle WSM data from the Oracle Database.
- The generated file describing the topology is *`ORACLE_HOME`*`/owsm/config/topology/topology.properties`.
- By default, only the most current version of each policy is inspected and described in the `topology.properties` file. To inspect earlier policy versions, use the `-Dinspect.level` parameter. By default, this is set to 0 to inspect only the current version. If you set it to 1, all version are inspected. Policies that are described in the `topology.properties` file can be configured for the destination installation. When policies are imported to the destination installation, they are imported with the configuration specified in the `topology.properties` file.

The -Dinspect.level parameter only specifies which policies are described in the topology.properties file. Regardless of this setting, all policy versions are exported from the source installation and imported to the destination installation. Policies that are *not* described in the topology.properties file, are imported with the configuration for the source installation. Once the cloning process is completed, you can use Web Services Manager Control to view and make changes to any of the policies.

- The wsmadmin inspectTopology command exports the Oracle Database tables to the following location on the source installation:
  - *ORACLE_HOME*/owsm/config/topology/export – Each database table is exported to an XML file in this directory.
  - *ORACLE_HOME*/owsm/config/topology/template – Several tables require some reformatting in order to be correctly imported into the destination database. The reformatted tables are in this directory.
- If you need to make any changes to the data in the database tables before it is imported into the destination installation, Oracle recommends that you copy the XML file from the *ORACLE_HOME*/owsm/config/topology/export directory to the *ORACLE_HOME*/owsm/config/topology/template directory, and make any changes to the copied file.
- During the cloning process, any files in the *ORACLE_HOME*/owsm/config/topology/template directory override the files in the *ORACLE_HOME*/owsm/config/topology/export directory.

Related Commands

applyTopology

initialize

This operation initializes the primary database.

Syntax

```
wsmadmin initialize
```

Usage

Use this operation to initialize and create the schema for an Oracle Lite database. However, to initialize an Oracle database, the schema must exist prior to using this operation. If you need to customize the data that is loaded into the database, use the dataloadConfigure and dataload operations. For more information, see "dataloadConfigure" on page C-11, and "dataload" on page C-11.

The initialize operation reads the following attributes from the coresv.properties file:

Table C–15 coresv.properties Used During dataload

Attribute Name	Description
dataload.messagelog.db.url	Location of the database URL.
dataload.messagelog.db.driver	Type of database driver, thick or thin.
dataload.messagelog.db.userid	User ID used to log in to the database.
dataload.messagelog.db.password	Encrypted password for logging in to the database.

Table C–15 (Cont.) coresv.properties Used During dataload

Attribute Name	Description
`dataload.db.import`	Value that imports only the data, or the schema and the data.

You will be prompted to provide a password to Oracle WSM Database. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

install

This operation is invoked by the installer automatically at the end of the installation procedure.

Syntax

```
wsmadmin install
```

Usage

The installer invokes this command at the end of the Oracle WSM configuration. If required, the configuration procedure includes the initialization of the bundled OC4J and Oracle Lite components. It also builds and deploys the J2EE applications, such as the Web Services Manager Control, Oracle WSM Policy Manager, and Oracle WSM Monitor. This single operation invokes and performs all the following operations:

- `configApps`
- `installOLite`
- `installOC4J`
- `initialize`
- `encodePasswords`

You will be prompted to provide passwords to Oracle WSM Database and Oracle Application Server. For the application server password, provide the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

installAgent

This operation builds and installs the J2EE agent based on the properties in the `agent.properties` file.

Syntax

```
wsmadmin installAgent
```

Usage

The `installAgent` operation requires the following attribute from the `agent.properties` file:

Table C–16 agent.properties Used During installAgent

Attribute Name	Description
`agent.componentType`	Type of agent can be either a client or server agent.

> **Note:** See Chapter 6, "Installing Oracle WSM Agents", for more information on setting properties for each type of agent.

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

installOC4J

This operation installs the bundled OC4J application server included with the standalone installation.

Syntax

```
wsmadmin installOC4J
```

Usage

This operation initializes the OC4J server ports for Oracle WSM. When this operation runs for the first time, it sets the OC4J administrator password.

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

installOLite

This operation installs the bundled Oracle Lite database included with the basic installation.

Syntax

```
wsmadmin installOLite
```

Usage

This operation performs the following functions:

- Reads the `install.db.*` attribute in the `install.properties` file
- Adds database entries to `%WinDir%/OCBC.ini`
- Creates and updates `%WinDir%/POLITE.ini`
- Creates a service for the Oracle Lite database, under the Windows Services Administrative Tool

You will be prompted to provide a password for the system user for the Oracle WSM Database (orawsm) instance. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

manageUsergroups

Use this operation to manage user groups in the database.

Syntax

```
wsmadmin manageUserGroups parameter
```

Parameters

- *parameter* – User or group name. The valid values are as follows:
  - `addUser`
  - `addGroup`
  - `addUserGroup`
  - `deleteUser`
  - `deleteGroup`
  - `deleteUserGroup`

Usage

The `manageUserGroups` operation requires certain attributes from the `manageUserGrops.properties` file. See the *Oracle Web Services Manager Administrator's Guide* for information about the changes you need to make to the properties file.

You will be prompted to provide a password to Oracle WSM Database. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

md5encode

Use this operation to encode a password associated with a user to the specified password file. This is used with file-based authentication.

Syntax

```
wsmadmin md5encode htpasswdfile username
```

Parameters

- *htpasswdfile* – Name of the file containing the user name and password
- *username* – User name in the text file

Usage

Run the md5encode command for each user name and password. You will be prompted to enter the password for the user specified at the command line. Each encrypted result gets appended to the end of the file. The following are examples in an htpasswdfile file:

```
johndoe:{MD5}JMnhX1KvxHwiW3V+e+4fnQ==
```

```
janedoe:{MD5}dqIXO+Y5M1TnL/pNbfEDCg==
```

migrate

This operation migrates the Oracle WSM 10.1.2 properties and database settings to Oracle WSM 10.1.3.1.0.

Syntax

```
wsmadmin migrate OWSM_10.1.2_dir
```

Parameters

- *OWSM_10.1.2_dir* – Directory where Oracle WSM 10.1.2 is installed

Usage

You will be prompted to provide passwords to Oracle WSM Database and Oracle Application Server. For the application server password, provide the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

For more information about the wsmadmin migrate command, see *Oracle Web Services Manager Upgrade Guide*.

purgePolicies

This operation deletes old versions of policies from Oracle WSM Policy Manager.

Syntax

wsmadmin purgePolicies *component_ID policy_version*

- *component_ID* – ID of the Web Services Manager Agent or Gateway
- *policy_version* – Version number of the policy. All older versions of policies (policies whose version number is less than *policy_version*) are purged.

Usage Notes

- If the version number specified with *policy_version* is greater than the current active policy, then all policies except the current active policy are purged. For example, if the current active policy has a version number of 5, and the wsmadmin purgePolicies is executed with *policy_version* set to 6, then versions 1 through 4 are purged. Version 5 is not purged and remains as the active policy.
- Policy version numbers are positive numbers. If a negative number is specified for the *policy_version,* no policies are purged.

start

This operation starts the OC4J application engine and the Oracle Lite database server included with the basic installation.

Syntax

```
wsmadmin start
```

Usage

The start operation reads the following attributes from the coresv.properties file:

Table C–17 coresv.properties Used During start

Attribute Name	Description
oc4j.j2ee.home	Location of the OC4J container.
proxy.host	Host name of the OC4J installation.
proxy.port	Port number of the proxy server.
noproxy.hosts	Names of the host machines that should not exchange data using the proxy.

Use this operation if you intend to use `startOLite` followed by `startOC4J`.

startOC4J

This operation starts the OC4J application engine included with the basic installation.

Syntax

```
wsmadmin startOC4J
```

Usage

The `startOC4J` operation reads the following attributes from the `coresv.properties` file:

Table C–18 coresv.properties Used During start

Attribute Name	Description
`oc4j.j2ee.home`	Location of the OC4J container.
`proxy.host`	Host name of the OC4J installation.
`proxy.port`	Port number of the proxy server.
`noproxy.hosts`	Names of the host machines that should not be accessed through the proxy. You can enter multiple host names, and use the vertical bar (|) character to separate them.

startOLite

This operation starts the Oracle Lite database server include in the basic installation.

Syntax

```
wsmadmin startOLite
```

stop

This operation stops the OC4J application engine and the Oracle Lite database server included with the basic installation.

Syntax

```
wsmadmin stop
```

Usage

This operation reads the `oc4j.j2ee.home` attribute in the `coresv.properties` file.

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

The `stop` operation combines the operations `stopOC4J` and `stopOLite`.

stopOC4J

This operation stops the OC4J application engine included with the basic installation.

Syntax

```
wsmadmin stopOC4J
```

Usage

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

stopOLite

This operation stops the Oracle Lite database server included with the basic installation.

Syntax

```
wsmadmin stopOLite
```

undeploy

This operation unbinds and undeploys the specified components from an OC4J instance.

Syntax

```
wsmadmin undeploy component
```

Parameters

- *`component`* – Name of the component being deployed. The valid values are as follows:
  - `all` – Deploys all components
  - `control` – Deploys the Web Services Manager Control
  - `gateway` – Deploys the Oracle WSM Gateway
  - `monitor` – Deploys the Oracle WSM Monitor
  - `policymanager` – Deploys the Oracle WSM Policy Manager

Usage

The `undeploy` operation reads the following attributes from the `install.properties` file:

Table C–19 coresv.properties Used During undeploy

Attribute Name	Description
`install.http.host`	Host name for the OC4J installation.
`install.oc4j.adminID`	Administrator's username for the OC4J installation.
`install.oc4j.instance`	Location of the OC4J instance.

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

uninstall

The installer invokes this operation automatically when uninstalling the Oracle WSM product.

Syntax

```
wsmadmin uninstall
```

Usage

The deconfiguration procedure includes uninstalling the OC4J and Oracle Lite components. It also uninstalls the J2EE applications. The operation will not uninstall the OC4J components if you do not supply the password. This operation performs the functions of all the following operations:

1. Runs `uninstallOC4J`. If you do not supply the password, OC4J is not uninstalled. You need to uninstall OC4J manually.
2. Runs `uninstallOLite`
3. Deletes configuration files

> **Note:** This operation does not clean up the database. You have to perform this step separately.

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

uninstallOC4J

This operation deactivates the OC4J application server included with the basic installation.

Syntax

```
wsmadmin uninstallOC4J
```

Usage

This operation performs the functions of the following operations:

- `undeployAll`
- `stopOC4J`

You will be prompted to provide a password to Oracle Application Server. Use the OC4J system administrator password. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

uninstallOLite

This operation uninstalls the Oracle Lite database server included with a Basic Oracle WSM installation.

Syntax

```
wsmadmin uninstallOLite
```

Usage

This operation performs the following tasks:

- Stops the Oracle Lite database server and removes the service from the system
- Removes the database entries from `%WinDir%/OCBC.ini`
- Clears `%WinDir%/POLITE.ini`

upgrade

This operation upgrades the Oracle WSM Database.

Syntax

```
wsmadmin upgrade database_type
```

Parameters

- *`database_type`* – Must be `oracle`, since this version of Oracle WSM only supports the Oracle database

Usage

You will be prompted to provide a password to Oracle WSM Database. For alternate ways to provide WSMADMIN passwords, see "Passwords for WSMADMIN Commands" on page C-3.

This version of Oracle WSM only supports the Oracle database.

D

Database Maintenance

This chapter explains how to connect the various supported databases to your Oracle Web Services Manager (Oracle WSM) deployment.

This chapter includes the following sections:

- How Oracle WSM Uses Databases
- Oracle Lite 10g
- Oracle 10g

How Oracle WSM Uses Databases

Oracle WSM Database stores information from the following Oracle WSM components:

- Oracle WSM Policy Manager
- Oracle WSM Monitor
- Oracle WSM Agents
- Oracle WSM Gateways

The Oracle WSM Database is also used to store the raw and compiled data associated with the Performance Metrics generated by the Oracle WSM Monitor.

Oracle Lite 10*g*

Oracle WSM comes bundled with Oracle Lite Database 10*g*. When you install Oracle WSM on Microsoft Windows as a standalone installation with the Basic Installation option, Oracle Lite 10*g* is automatically installed.

If the installer confirms that the property value of the parameter `install.db.type` has been set to `oracle`, then it installs Oracle WSM and connects it to the specified database.

The `wsmadmin initialize` operation automatically creates the database tables required by Oracle Web Services Manager. It then uploads the requisite data to those tables. For details, see "initialize" on page C-21.

> **Note:** Oracle Lite is optimal for demonstration and proof-of-concept deployments of Oracle WSM on a single host machine. It is not certified for production environments. You must integrate production Oracle WSM deployments with Oracle 10*g*.

Uninstalling Oracle Lite

The following procedure describes how to uninstall Oracle Lite.

To uninstall the bundled Oracle Lite instance

1. Open a command window, and go to the directory where you installed Oracle Lite.

   The default location of the Oracle Lite installation is indicated by the value of the `olite.install.home` property in the `install.properties` file.

2. From the command prompt, run the following command:

   ```
   wsmadmin uninstallOLite
   ```

   For more information about uninstalling Oracle Lite using the `wsmadmin` command, see "uninstallOLite" on page C-28.

Oracle 10*g*

Integration of Oracle 10*g* with Oracle Web Services Manager requires completion of the following tasks:

1. Modify the `install.properties` file prior to installing Oracle Web Services Manager, setting the database properties according to Table D–1.

Table D–1 install.properties Settings for Oracle 10g

Property	Description
`db.type`	Specify `oracle`
`db.driver.type`	Specify `thick` or `thin`. The JDBC thin driver allows you to use 100% Java to access Oracle data. The thick driver uses Oracle-specific native code (that is non-Java) which provides a performance boost compared to the thin driver at the cost of significantly larger size and client-side installation.
`db.host`	The machine name, IP address, and port of the computer hosting the database. For a single node, this is specified using the syntax `host:port` For a RAC database, this is specified using the syntax `host1:port1^host2:port2^`
`db.name`	The service name of the database instance in which the Oracle Web Services Manager Registry resides (the default is CCORE).
`db.userid`	The schema or account used to log in to the database.
`db.password`	The password associated with the account specified by `db.userid`.
`db.type`	Specify `oracle`

2. Initialize the Oracle database by opening a command window, navigating to *`ORACLE_HOME`*`\owsm\bin` and executing the following command:

   ```
   wsmadmin initialize
   ```

   For more information about using the `wsmadmin initialize` command, see "initialize" on page C-21.

Changing Passwords

You may want to change passwords for security reasons; in particular, database passwords may need to be changed on a regular basis. Oracle recommends that you use a single Oracle WSM Database in your Oracle WSM environment. However, Oracle WSM components can support multiple instances of the database.

The following procedure describes how to change a database password for a single instance of an Oracle WSM Database.

To change a password for a single database instance

1. Change the password for the OWSM schema in the Oracle Database.
2. Restart the Oracle Application Server on which the Oracle WSM components are deployed.
3. Edit the `ORACLE_HOME\owsm\bin\coresv.properties` file, and update the `dataload.messagelog.db.password` property with the new database password.
4. Encode the password in the `coresv.properties` file.

   ```
   wsmadmin encodePasswords coresv.properties dataload.messagelog.db.password
   ```

5. Copy the database configuration properties to the configuration directory property files.

   ```
   wsmadmin copyDBConfig
   ```

6. Redeploy the Oracle WSM applications either individually or using the `all` parameter.

   ```
   wsmadmin deploy oc4j_admin_password all
   ```

7. Restart the Oracle Application Server instance.

Complete the procedure in "Updating the Database Configuration Properties" on page D-4."

The following procedure describes how to change a database password for multiple instances of an Oracle WSM Database.

To update and reconcile a password for multiple database instances

1. Change the password for the OWSM schema in the Oracle Database.
2. Restart the Oracle Application Server on which the Oracle WSM components are deployed.
3. Update the relevant attribute in the application properties file for each application. These files are located in *`ORACLE_HOME`*`\owsm\config\`*`application_name`*.
4. Encode the password for each property that was changed in step 3.

   ```
   wsmadmin encodePasswords file_name property_name
   ```

5. Redeploy the Oracle WSM components that connect to the database.

   ```
   wsmadmin deploy oc4j_admin_password component
   ```

6. Restart the Oracle Application Server instance.
7. Repeat this procedure for each Oracle Database that is used by an Oracle WSM component.

Complete the procedure in "Updating the Database Configuration Properties" on page D-4."

Updating the Database Configuration Properties

When you update the database password for Oracle WSM components, the database configuration properties for the Oracle WSM Agents and Gateways do not get updated. Therefore, you will not be able to access the message logs for these components. The workaround for this problem is to manually edit the properties for the components with the new password.

1. Log in to Web Services Manager Control.
2. In the navigation pane, select **Policy Management.**
3. Click **Edit** for the Oracle WSM Agent or Gateway.
4. Edit the `cfluent.messagelog.db.password` with the new database password, and save this change.

E

Authentication Sources

This chapter describes the authentication types supported by Oracle Web Services Manager (Oracle WSM) and how to configure them.

This chapter includes the following sections:

- Overview of Authentication
- Oracle Access Manager
- Active Directory
- Using an LDAP Directory to Authenticate Users

Overview of Authentication

Oracle WSM supports two types of authentication:

- **System Authentication –** System authentication is the verification of the identities of the Oracle WSM administrator and any other users authorized to manage system components and features. By default, users and passwords are stored in the administrative registry that resides in the Oracle WSM Database. You can also configure Oracle WSM to authenticate system users using an LDAP v3 directory. See *Oracle Web Services Manager Administrator's Guide* for information on roles.
- **End User Authentication –** End-user authentication verifies the identity of users who request services that are protected by Oracle WSM. When a requesting application requests a connection, the Policy Enforcement Point (PEP) that enforces the authentication and authorization policy steps for the Oracle WSM Policy Manager handles the request. For an overview of PEPs, see "Oracle Web Services Manager Deployment" on page 1-1. You can use any of the following resources to enforce the authentication and authorization policy steps to verify the identity of the requesting application:
  - Oracle Access Manager
  - CA eTrust SiteMinder
  - Standard LDAP v3 directories
  - Microsoft Active Directory

Oracle Access Manager

Oracle Access Manager and Oracle WSM provide an integrated and comprehensive identity management, Web services security, and system monitoring solution.

This section explains how to integrate Oracle WSM and Oracle Access Manager to authenticate users and to verify their privileges.

Note: Oracle Access Manager was previously known as Oblix NetPoint and Oracle COREid.

Oracle Access Manager Integration Overview

You can use the Oracle Access Manager to provide authentication and authorization services for Oracle WSM operations. Oracle Access Manager authenticates a client's identity and then authorizes different levels of access depending on the identity of the client.

This overview briefly describes the components of Oracle Access Manager. It also identifies the requirements for integrating Oracle Access Manager with Oracle WSM.

Oracle Access Manager is comprised of the following components:

Identity System – The Identity System manages identity information about individuals, groups, organizations, and other objects. The Identity System also leverages user identity and policy information for other applications and systems across the enterprise. This eliminates the need to create and manage separate user identity repositories for each application.

The Identity System consists of an Identity Server and a WebPass component. Administrators configure these components using a Web-based administration tool known as the Identity System Console.

See the *Oracle Access Manager Installation Guide* and the *Oracle Access Manager Identity and Common Administration Guide*.

Access System – The Access System is the access-control system that provides single sign-on across any Web application. It supports a variety of access policies, and is fully integrated with the Identity System so that changes in user profiles are instantly reflected in the Access System’s policy enforcement.

The Access System consists of the Policy Manager, Access Server, and WebGate:

- **Policy Manager –** The Policy Manager provides a Web-based interface where administrators can create and manage access policies. The Policy Manager also communicates with the directory server to write policy data, and communicates with the Access Server when certain policy modifications are made.

  For more information about installing the Policy Manager, see *Oracle Access Manager Installation Guide*.

  Master Access Administrators and Delegated Access Administrators use the Policy Manager and Access System Console.

- **Access System Console –** This is a Web-based application that provides administrators with the features and functionality related to System Configuration, System Management, and Access System Configuration.

- **Access Server –** Access Server receives requests and then queries authentication, authorization, and auditing rules in the directory server. Based on the information in the directory server, the Access Server sends the authentication scheme, validates user credentials, authorizes the user, audits, and manages the session.

  For more information and an overview of Access Server processes, see *Oracle Access Manager Introduction*.

- **WebGate –** WebGate is a server plug-in Oracle Access Manager access client that intercepts HTTP requests for Web resources and forwards them to the Access Server for authentication and authorization. A WebGate is shipped out-of-the-box with Oracle Access Manager.
- **AccessGate** – An AccessGate is a *custom* access client that is specifically developed using the Software Development Kit (SDK) and Oracle Access Manager APIs, either by the customer or by Oracle. An AccessGate is a form of access client that processes requests for Web and non-Web resources (that is, non-HTTP) from users or applications.

Integration Procedure and Requirements

The following components must be installed in the order specified in the procedure that follows. For more information, see *Oracle Access Manager Installation Guide.*

Task Overview: Preparing for Integration

1. Install Oracle Web Services Manager, as described in *Oracle Web Services Manager Installation Guide*.
2. Install and set up the Identity, System including:
    a. Identity Server
    b. WebPass

    See *Oracle Access Manager Installation Guide* for more information.
3. Install and set up at least one instance of each of the following components:
    a. Policy Manager (includes the Access System Console):
        - Define the policy base during the Policy Manager setup.
        - Define the policy domain root during the Policy Manager setup.
        - Accept the default authentication schemes during the Policy Manager setup. (Otherwise, you must create the authentication schemes using the Access System Console after setup.)
        - Create the Master Access Administrator who will have the authority to create policy domains, resource types, access control templates called schemes, and to assign other administrators the role of Delegated Administrator of a policy domain.

    b. Access Server.
    c. Create AccessGate and install it on the same machine as Oracle WSM.

    See the *Oracle Access Manager Developer Guide* for more information.
4. Using the Oracle Access Manager Policy Manager, protect resources:
    a. Create a policy domain
    b. Bind the resource types to URL mappings.
    c. Create one or more authorization rules and associate users and groups with these rules. An authorization rule identifies who can access a resource and who is explicitly denied access to a resource. You can include one or more authorization rules in an authorization expression for a policy domain or policy. See the "Configuring User Authorization" chapter in Oracle Access Manager Access System Administration Guide.

d. Create default rules, including an authentication rule, authorization expressions, and an audit rule for the policy domain.

e. Create policies to protect subsets of resources in the policy domain. Policies enable you to differentiate how subsets of resources in a domain are protected. You can use policies to establish more or less stringent protection for a subgroup of resources of a policy domain. See "Protecting Resources with Policy Domains" in Oracle Access Manager Access System Administration Guide

f. Test the policy domain.

See *Oracle Access Manager Access System Administration Guide* for more information.

5. Configure Oracle WSM to use AccessGate.

   See "Configuring Oracle WSM to Use a Custom AccessGate" on page E-5.

6. Configure AccessGate to work with Oracle WSM.

   See "Configuring AccessGate to Work with Oracle Access Manager" on page E-5.

7. Configure policy steps in Oracle WSM.

   See "Configuring Policy Steps in Oracle WSM" on page E-8.

The next sections describe authentication mechanisms, resources, and URL patterns for the Access Server and AccessGate.

Authentication Mechanisms

Oracle WSM supports the following Oracle Access Manager authentication mechanisms:

- Oracle Access and Identity Basic Over LDAP (formerly COREid Basic Over LDAP)
- Basic over LDAP
- Client Certificate

Oracle WSM collapses these three authentication methods into two methods and implements them as User Name and Password and Client Certificates.

Table E–1 shows the correspondence between the Oracle WSM and Oracle Access Manager authentication mechanisms.

Table E–1 Authentication Mechanisms Compared

Oracle Web Services Manager	Oracle Access Manager
User Name and Password	Basic over LDAP
User Name and Password	Oracle Access and Identity
Client Certificates	Client Certificates

When you integrate Oracle WSM and Oracle Access Manager, you must decide whether to use user names and passwords (Basic over LDAP) or client certificates for authentication.

Resources

The only resource supported is HTTP.

URL Patterns

URL patterns must look like the following example:

```
http:///gateway/services/TimeService
```

Configuring Oracle WSM to Use a Custom AccessGate

Complete the following steps to set up authentication policies in the Oracle WSM Policy Manager:

- Install Oracle WSM.
- Install AccessGate on the system where Oracle WSM resides.
- Change the startup script for Oracle WSM to include the library path for AccessGate Java native libraries.

Microsoft Windows

For a Microsoft Windows installation, this procedure assumes that you have installed the Access Manager SDK in the following directory

```
C:\Oblix\NetPoint\AccessServerSDK
```

The Oracle WSM startup script is the following:

```
install_home\bin\wsmadmin.bat
```

where *install_home* is the root directory for your Oracle WSM installation.

Linux

For a Linux installation, this procedure assumes that you have installed the Access Manager SDK in the following directory:

```
/Oblix/NetPoint/AccessServerSDK
```

The Oracle WSM startup script is the following:

```
install_home\bin\wsadmin.sh
```

where *install_home* is the root directory for your Oracle WSM installation.

Configuring AccessGate to Work with Oracle Access Manager

To facilitate the handshake between AccessGate and Oracle Access Manager, run the utility configureAccessGate.exe, located in the following directory:

```
installdirectory\AccessServerSDK\oblix\tools\configureaccessgate
```

where *installdirectory* is the root folder for your Access Manager SDK installation.

From a command prompt, execute the following command:

```
configureAccessGate.exe -i installdirectory\AccessServerSDK -t AccessGate -P
AccessGatePwd -w CoreSvAccessGt -m open -h AccessServerHostname -p
accessserverport -a accessserverid -r AccessServerpassphrase
```

The parameters are described in Table E–2.

Table E–2 Description of configureAccessGate.exe Parameters

Parameter	Description
`-i` *`installdirectory`*	Installation directory for AccessServerSDK
`-t` *`AccessGate`*	AccessGate keyword. Enter as shown.
`-P` *`AccessGatePwd`*	AccessGate password. When an entry for AccessGate is created in Oracle Access Manager, a password may be specified.
`-w` *`CoreSvAccessGt`*	AccessGate name that was specified when the AccessGate entry is created in Oracle Access Manager.
`-m open`	Oracle Access Manager intercomponent mode. The valid values are `open, simple,` or `cert.`
`-h` *`AccessServerHostname`*	Name of the host on which Access Server is installed.
`-p` *`accessserverport`*	Port on which Access Server is running.
`-a` *`accessserverid`*	Access Server name in Oracle Access Manager
`-r` *`AccessServerpassphrase`*	Access Server simple mode password. This password is required if Access Server in Oracle Access Manager is running in `simple` mode. It is not required if Access Server is running in `open` mode.

Creating Policies Using the Oracle Access Manager Policy Manager

The following is a summary of tasks that you must complete to set up policies on an Oracle Access Manager Policy Manager.

Task overview: Setting up a Policy in the Oracle Access Manager Policy Manager

1. Create a policy domain.
2. Create resource-type-to-URL mappings.
3. Create policies in the policy domain.
4. Define default authentication rules.
5. Define default authorization rules and associate users and groups with the rules.

Sample Policy Creation: Oracle Access and Identity Basic Over LDAP

The following sections illustrate how to use the Oracle Access Manager Policy Manager application in Oracle Access Manager to create a policy domain named Oracle WSM. It also describes how to use an Oracle Access and Identity Basic Over LDAP authentication scheme to protect a service named *TimeService*. This service was created in the Oracle Web Services Manager Gateway (gateway).

The following example shows the service ID for TimeService is SID0002001.

To protect the TimeService resource using an Oracle Access and Identity authentication scheme

1. Launch the Access System Console.

   Enter the following URL in a browser:

   ```
   http://WebPass_hostname:port/access/oblix
   ```

where `WebPass_hostname` refers to the machine hosting the WebPass application server, `port` refers to the HTTP port number of the WebPass application server instance, and `/access/oblix` connects to the Access System Console.

2. Select the Oracle Access Manager Policy Manager application, and click **Create Policy Domain** in the left navigation pane.
3. Enter Oracle WSM in the Name field, specify an optional description, and click **Save.**
4. To add resources, click the **Resources** tab, then click **Add.**
5. Enter the following details for the gateway/services/TimeService and gateway/services/SID0002001 URLs:
   - **Resource Type –** http
   - **URL Prefix –** /gateway/services/TimeService
6. To add an Authorization rule for this Policy Domain, click the **Authorization Rules** tab, then click **Add**.
7. Enter and save the following information:
   - **Name –** SimpleAuthRule
   - **Enabled –** Yes
   - **Allow takes precedence –** No
8. Click the **Allow Access** and complete the following steps:
   a. People – Click **Select User** to select by user name. Use the Search facility to display configured users, and click Add before the name of each user who is allowed to access resources protected by this rule.
   b. Role – Select **No Role** in the Role selection box to prevent users from being selected based on roles or select Anyone to allow anyone access to the protected resources.
   c. Rule – Enter an LDAP filter that specifies the users and groups who are allowed to access
9. Click **Select User,** and add the users to whom you want to give access.
10. Click **Policies** and create a policy named *TimePolicy* with the following information:

    Name – TimePolicy
    Resource Type – http
    Resource – all
    Resource operations – Get, Post, Other

Sample Policy Creation: Client Certificates

The following paragraphs describe how to use the Oracle Access Manager Policy Manager to create a policy domain called *Oracle WSM*. It shows how to use client certificates to protect the TimeService service.

To create a policy using a client certificate authentication scheme

1. Launch the Access System Console and select the **Policy Manager.**
2. From the Policy Manager, click **Create Policy Domain** in the left navigation pane.
3. Enter Oracle WSM in the Name field and click **Save.**

4. In the Description field, type a brief description of the policy domain.
5. Click the **Resources** tab, and add the following resources:

   Resource Type – http
   URL Prefix – /gateway/services/TimeService
   Resource Type – http
   URL Prefix – /gateway/services/SID0003001

6. Click the **Default Rules** tab, select the Authentication Rule sub-tab, click **Add**, give the rule a name, and choose the Client Certificates authentication scheme.
7. Click the **Authorization Rules** tab, click **Add**, then add the following information:

   Name – SimpleAuthRule
   Enabled – Yes
   Allow takes precedence – No

8. Save the rule.
9. Click **Allow Access** sub-tab, and select the allowed users.
10. Click **Select User** and add the users to whom you want to give access.
11. Click **Policies,** then create TimePolicy by entering the following information:

    Name – TimePolicy
    Resource Type – http
    Resource – all

12. Save the policy rule.
13. Click the policy rule that was just created (TimePolicy).
14. Click Authentication Rule sub-tab.

The following sections describe these tasks in detail.

Configuring Policy Steps in Oracle WSM

The following sections illustrate how to configure policy steps for a gateway:

- Enforcement with User Name and Password
- Enforcement with Certificates

Enforcement with User Name and Password

The following example shows how to configure policy steps for User Name and Password for a gateway.

1. Register the TimeService service to a gateway.
2. Modify the request pipeline for this service and include the following steps in the following order:
   - Extract Credentials
   - Oracle Access Manager Authenticate Authorize step

     Namespaces – Enter a list of name spaces separated by spaces (white space-delimited strings).

     UserID xpath – Enter an xpath that points to the location of the user name, for example, wsse:Username.

Password xpath – Enter an xpath for password, for example, wsse:Password.

AccessGate Install Directory – The following is the default installation directory for AccessGate:

- On Microsoft Windows, the default directory is `C:\Program Files\Netpoint\AccessServerSDK`.
- On Linux, the default directory is: `/opt/netpoint/AccessServerSDK`.

These directory paths differ if your AccessGate installation directory is different.

3. Save the policy.
4. Commit the changes.
5. Send a request to the gateway with the following service URL, including the user name and password in the HTTP header.

   ```
   http://<host:port>/gateway/services/TimeService
   ```

   The user name and password be for one of the users allowed by the current Oracle Access Manager authorization rules.
6. When the policy steps correctly have been configured correctly, the service will respond. Otherwise a SOAPFault error message is returned.

Enforcement with Certificates

1. Register the TimeService service on a gateway.
2. Modify the request pipeline for this service and include the following steps in the following order:
   - Verify Signature
   - Oracle Access Manager Authenticate Authorize step

     Keystore location – Enter the location of the keystore file.

     Keystore password – Enter the password for the keystore file.

     Signer's public-key alias – Enter the signer's public-key alias.

     AccessGate Install Directory – The first of the following two examples is for Windows, and the second is for Linux.

     ```
     C:\Oblix\NetPoint\AccessServerSDK
     /Oblix/NetPoint/AccessServerSDK
     ```

     These directory paths differ if your AccessGate installation directory is different.
3. Save the policy by clicking **Save.**
4. Commit the changes by clicking **Commit.**
5. Send a request to the gateway through the following URL:

   ```
   http://<host:port>/gateway/services/TimeService
   ```
6. If the policy steps have been configured correctly, the service will respond. Otherwise, a SOAPFault error message is returned.

Oracle Access Manager Authenticate Authorize Configuration

The Oracle Access Manager Authenticate Authorize policy step uses the Java Native Interface (JNI) libraries. Therefore, you must configure your environment variables to load the shared libraries.

OPMN-Managed Deployments

If your system is managed using Oracle Process Manager and Notification (OPMN), then use the following procedure for your platform. Your Oracle WSM installation is managed using OPMN if you use the `opmnctl start` and `stop` commands to start and stop Oracle WSM.

1. Open the *ORACLE_HOME*/opmn/conf/opmn.xml file.
2. Find the OC4J instance that is hosting the gateway or agent component. Look for the following entries in the opmn.xml file.

```
<ias-component id="group1" status="enabled">
     <process-type id="Name_of_OC4J_Instance" module-id="OC4J"
STATUS="ENABLED">
       <module-data>
         <category id="start-parameters">
           <data id="java-options" value="server  ..."/>
```

3. On Microsoft Windows, add the path for the Access Server SDK to the Java options as shown below in bold:

```
<ias-component id="group1" status="enabled">
     <process-type id="Name_of_OC4J_Instance" module-id="OC4J"
status="enabled">
       <module-data>
         <category id="start-parameters">
           <data id="java-options" value="server
             -Djava.library.path=c:\pathto\AccessServerSDK\oblix\lib ..."/>
```

4. On Linux, add the `LD_LIBRARY_PATH` and `LD_ASSUME_KERNEL` environment variables as shown in bold:

```
<ias-component id="group1" status="enabled">
   <environment>
      <variable id="LD_LIBRARY_PATH" value="/pathto/AccessServerSDK/oblix/lib"
append="true"/>
      <variable id="LD_ASSUME_KERNEL" value=2.4.19"/>
    </environment>
     <process-type id="Name_of_OC4J_Instance" module-id="OC4J"
status="enabled">
       <module-data>
         <category id="start-parameters">
           <data id="java-options" value="server
              ..."/>
```

Standalone Oracle WSM Basic Install Deployments

If you installed a standalone version of Oracle WSM using the Basic installation option, then your Oracle WSM is not managed by OPMN. Oracle WSM is started using the `wsmadmin` command. If this is the case, then perform the step appropriate to the platform on which Oracle WSM is installed:

- On Microsoft Windows, edit the *ORACLE_HOME*\owsm\bin\wsmadmin.bat file. Add \pathto\AccessServerSDK\oblix\lib to the `Path` environment variable.

- On Linux, edit the *ORACLE_HOME*/owsm/bin/wsmadmin.sh file. Add /pathto/AccessServerSDK/oblix/lib to the `Path` environment variable, and set the LD_ASSUME_KERNEL environment variable to 2.4.19.

Active Directory

You can configure Active Directory to support both Oracle WSM System User authentication and End-User authentication.

Configuring Active Directory for End User Authentication

To configure Active Directory to provide authentication services for end users, you must verify that your PEP supports the Active Directory Authenticate policy step, then add that step to the policy associated with the PEP. For more information, see *Oracle Web Services Manager Administrator's Guide*.

Using an LDAP Directory to Authenticate Users

You can configure an LDAP Directory to support either Oracle WSM System User authentication, or end-user authentication, or both. For more information, see the "Managing Oracle Web Services Manager Roles" chapter in *Oracle Web Services Manager Administrator's Guide*.

F

Updating Host Names

This appendix describes the configuration changes you must make if the hosts on which your Oracle Web Services Manager (Oracle WSM) components reside change.

Updating Host Names

To dynamically generate some URLs, Oracle WSM needs to know the correct host names. If your host names change, you must update the `install.properties` file as described in the following procedures.

To change a host name for a single-host deployment

Follow this procedure if all Oracle WSM components are installed on one host.

1. Update the `install.http.host` property in the Database Connections section of the *`ORACLE_HOME`*`/owsm/bin/install.properties` file.
2. From the command prompt, execute the `wsmadmin configApps` command:

   ```
   wsmadmin configApps
   ```

3. From the command prompt, execute the `wsmadmin deploy` command:

   ```
   wsmadmin deploy component
   ```

See Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information.

To change a host name for a multiple-hosts deployment

Follow this procedure if your Oracle WSM components are distributed across multiple hosts.

1. Update the attribute in the application properties file for each application. These are located in *`ORACLE_HOME`*`/owsm/config`.
2. Update and reconcile the host names for all applications affected by changes to the HTTP settings.
3. From the command prompt, execute the `wsmadmin deploy` command:

   ```
   wsmadmin deploy all
   ```

4. Repeat this procedure for all of the HTTP ports in your deployment.

See Appendix C, "Oracle Web Services Manager WSMADMIN Commands" for more information.

Index