Oracle® Web Services Manager

Administrator's Guide

10*g* (10.1.3.3.0)

E10299-01

June 2007

ORACLE®

Oracle Web Services Manager Administrator's Guide, 10*g* (10.1.3.3.0)

E10299-01

Primary Author: Laureen Asato

Contents

4 Content Routing with Oracle Web Services Manager Gateways

5 Managing Oracle Web Services Manager Policies

6 Monitoring Oracle Web Services Manager

10 Troubleshooting

A Oracle Web Services Manager Policy Steps

B Oracle Web Services Manager Test Page

List of Figures

List of Tables

Preface

Oracle Web Services Manager (Oracle WSM) is designed to ease the installation, configuration, and management of Web services across a wide range of deployment environments.

This guide provides system administrators with information on how to use Oracle Web Services Manager Gateways and Oracle Web Services Manager Agents to secure Web Services, and how to monitor and manage the Oracle Web Services Manager environment.

This Preface includes the following sections:

Audience

This guide is for Oracle WSM system administrators who are responsible for administering Oracle WSM at a particular site. Oracle WSM system administrators may be responsible for managing policies for Oracle WSM Gateways and Oracle WSM Agents, assigning roles to users of the system, and monitoring the Oracle WSM environment.

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.

Related Documents

For more information, see the following documents in the Oracle Web Services Manager 10*g* (10.1.3.3.0) documentation set:

- *Oracle Web Services Manager Extensibility Guide*
- *Oracle Web Services Manager Deployment Guide*
- *Oracle Web Services Manager Quick Start Guide*
- *Oracle Web Services Manager Installation Guide*

Conventions

The following text conventions are used in this guide:

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.

File Path Locations

When describing the location of files in this guide, 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 Microsoft Windows operating system, replace the forward slashes with back slashes (\). For example:

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

1

Getting Started with Oracle Web Services Manager

This chapter provides an introduction to Oracle Web Services Manager (Oracle WSM) software and an overview of Oracle WSM product components.

This chapter includes the following sections:

Starting the Oracle WSM Server

After you have completed installing Oracle WSM, start the Oracle WSM server. For the standalone Oracle WSM installation, the Oracle WSM server should be started using the `wsmadmin` command-line tool.

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

On Windows

- Open a command window and execute the following command:

```
C:\> ORACLE_HOME\owsm\bin\wsmadmin start
```

On UNIX Platforms

- Open a command window and execute the following command:

```
$ ORACLE_HOME/owsm/bin/wsmadmin.sh start
```

Once the server is started, you can open Web Services Manager Control and use it to create and register your **policy** enforcement points, create policies, and manage and monitor your system.

> **Note:** If you have installed Oracle WSM as part of the Oracle Application Server 10*g* Release 3 (10.1.3.1.0) release, also known as the Oracle SOA Suite (10.1.3.1.0), the Oracle WSM Server is automatically started after installation. Refer to *Oracle Application Server Administrator's Guide* for more information on starting the server.

Web Services Manager Control

Following the initial Oracle WSM installation, you can access Oracle WSM Policy Manager operations and functions through the Oracle Enterprise Manager 10*g* Web Services Manager Control (Web Services Manager Control), which lets you configure and update Oracle WSM component **gateway**s and agents as well as register Web services to manage with Oracle WSM. You can also use Web Services Manager Control to define and update operational policies and **step**s, which you can then upload to configured gateways and agents.

Similarly, you can view **metric**s collected by Oracle WSM Monitor and monitor the status and operation of Oracle WSM managed services from Web Services Manager Control. Web Services Manager Control lets you perform a number of other operations as well. For example, an Administration section lets the Oracle WSM system administrator assign or delegate administrative or support permissions to other users for specific gateways, agents, or managed services.

In addition, Web Services Manager Control provides access to operations management functions that let you define conditions and options for generating alerts or alarms, and sending notifications when those alarms are generated.

Web Services Manager Control allows you to control access and permissions of other users to perform Oracle WSM operations by assigning or mapping various user groups already defined in your environment to Oracle WSM roles. By assigning Oracle WSM roles to groups (defined and stored in the Oracle WSM Database or maintained in your own LDAP (Lightweight Directory Access Protocol) store, you can then choose which groups are authorized to administer individual Oracle WSM components and individual Web services managed by Oracle WSM. (See Chapter 7, "Managing Oracle Web Services Manager Roles" for more information on managing Oracle WSM access and permissions.)

Accessing Web Services Manager Control

To access Oracle Web Services Manager

Open a Web browser and enter the following URL:

`http://host_name:port/ccore`

- If Oracle Web Services manager is installed as a standalone product using the Basic installation option, then *host_name* is the fully qualified name of the host where Oracle Web Services Manager is installed, and *port* is the **HTTP port** on which the server is listening. By default, this port is 3115. Enter `ccore` as shown in the URL.
- For all other installations, *host_name* is the fully qualified name of host on which the HTTP listener resides or the name of the load balancer that sits in front of the Oracle Web Services Manager installation, and *port* is the HTTP port on which the server is listening. Enter `ccore` as shown in the URL.

For example:

```
http://jdoe-pc.us.oracle.com:8888/ccore
```

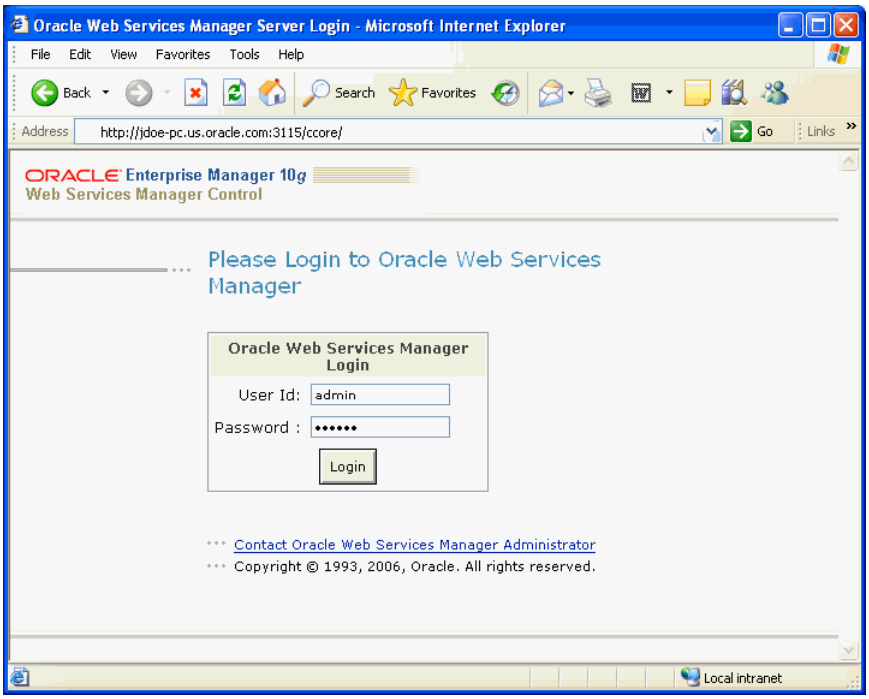
To log in to Oracle Web Services Manager

The credentials you use to log in to Oracle Web Services Manager will differ depending on whether or not JSSO is enabled. If Oracle Web Services Manager is installed as part of the Oracle Application Server 10*g* Release 3 (10.1.3.1.0) (also known

as Oracle SOA Suite) using the Basic install option, then JSSO is enabled by default. For all other installations, JSSO is not enabled by default.

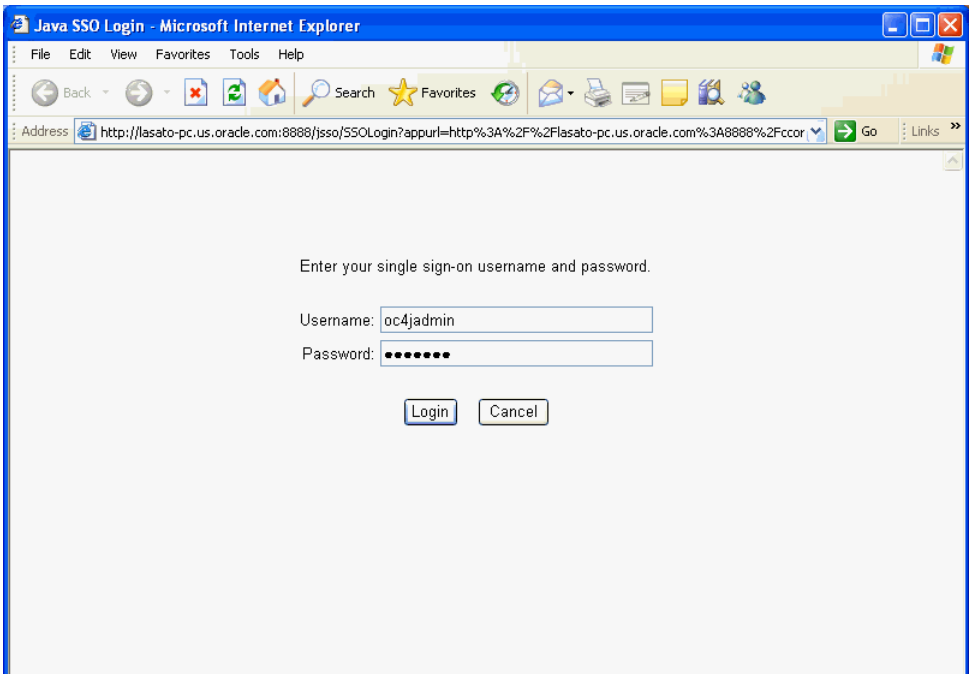
- **If JSSO is not enabled** – Oracle WSM displays a login page (Figure 1–1). Enter a user name and password for the Oracle WSM administrator. By default the user name is *admin* and the password is *oracle*.

Figure 1–1 Oracle Web Services Manager Login Page



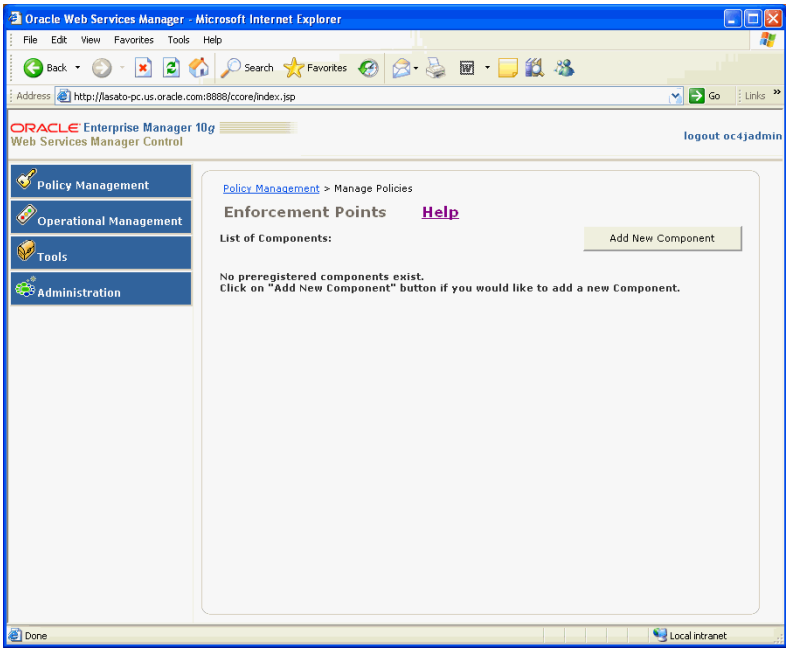
- **If JSSO is enabled** – Oracle WSM displays a different login page, one that looks like Figure 1–2. Log in using the OC4J (Oracle Containers for J2EE) administrator user name and password.

Figure 1–2 Oracle Web Services Manager Login Page



After logging in to Web Services Manager Control, the following page is displayed (Figure 1–3).

Figure 1–3 Web Services Manager Control



Web Services Manager Control Menus

There are four menus in the navigation pane of Web Services Manager Control:

- **Policy Management** – Use this menu to configure and update Oracle WSM gateways and agents, register and update services, and add or update policy templates.
- **Operational Management** – Use this menu to monitor the status and **performance** of Oracle WSM components and managed **Web service**s, create custom views of Oracle WSM metrics, and create and view rules for alarm generation and **notification**.
- **Tools** – Use this menu to conduct ping and test operations of Web services and Oracle WSM components, perform operations provided by individual services, and view **SOAP message** requests and responses.
- **Administration** – Use this menu to manage Oracle WSM access and permissions by assigning Oracle WSM roles to user groups.

When you click one of the main menu options, Web Services Manager Control displays additional menu choices.

When users log in, Oracle WSM automatically determines a user's permissions and the Oracle WSM components, managed Web services, and Web Services Manager Control operations the user is allowed to access. For example, the GatewayAdmin group is assigned to an Oracle WSM Administrator **role** and grants access to the Oracle WSM **gateway** component. A user assigned to the Gateway Admin group is allowed to view and perform any administration operations for that gateway and associated services. At the same time, other users who are members of groups assigned to Oracle WSM roles with fewer permissions are restricted in the operations they are allowed to perform on the same gateway and associated services.

Note: See Chapter 7, "Managing Oracle Web Services Manager Roles" for more information on assigning Oracle WSM roles to user groups.

2

Registering Web Services to an Oracle Web Services Manager Gateway

This chapter includes the following sections:

Registering Web Services to an Oracle Web Services Manager Gateway

Before you can create and edit policies for an Oracle Web Services Manager Gateway, you must first register the desired Web services. The following information provides general directions for registering Web services to a gateway.

You can register a **service** by specifying the Web Services Definition Language (**WSDL**) URL or by importing the service. Refer to "Adding a Service to an Oracle Web Services Manager Gateway" on page 2-2, for information on registering a service by specifying the WSDL URL. For information on registering a service by importing it, refer to "Discovering Web Services" on page 3-2.

Proxy Settings

If you import a **Web service** by specifying an external URL, you must set your HTTP proxy settings. For the standalone Oracle WSM installation, this is set in the *OWSM_HOME*/owsm/bin/coresv.properties file. If you installed Oracle WSM as part of the Oracle Application Server 10*g* Release 3 (10.1.3.1.0), also known as Oracle SOA Suite (10.1.3.1.0), set your proxy settings in the *ORACLE_HOME*/opmn/conf/ompn.xml file. See "Error When Importing WSIL" on page 10-2 for more information.

Registering WSDLs from SSL-Enabled Sites

If you are importing WSDLs from sites enabled for Secure Sockets Layer (SSL), that is, the WSDL uses an HTTPS URL, then you must enable SSL when you start the Oracle Web Services Manager server. Add SSL system attributes to the oc4j.start target in the following file: *ORACLE_HOME*/owsm/scripts/oc4j.xml.

Accessing Virtualized Web Services

You can access a virtualized Web service either by using the service identifier (ID) or the Web service name. Therefore, either of the following URLs could be used to access the TimeService Web service whose service ID is SID0003006:

- `http://jdoe.us.oracle.com:3115/gateway/services/SID0003006?wsdl`
- `http://jdoe.us.oracle.com:3115/gateway/services/TimeService?wsdl`

The naming conventions for URLs apply.

Adding a Service to an Oracle Web Services Manager Gateway

To add a Web service to a gateway

1. Select **Policy Management**, then select **Register Services** from Web Services Manager Control.

   Web Services Manager Control displays a list of the currently registered gateways.

2. Click **Services** for the gateway for which you want to register Web services.

   Web Services Manager Control displays a list of the Web services currently registered to the gateway.

3. Click **Add New Service**.

4. Provide the details for the new service. Click **Help** to get information on the fields.

5. Click **Next**.

   Web Services Manager Control displays the Configure Messenger Step for New Service page where you configure the Web service transport protocol (also known as the outgoing transport protocol) you selected. See "Oracle WSM Transport Protocols" on page 2-3 for more information.

6. Configure the fields as required.

   Click the question mark (?) for help on the protocol parameters. This information is also provided in "Configuring the Outgoing Transport Protocol" on page 2-4.

7. Click **Finish**.

   The Services confirmation message is displayed.

8. Click **OK**.

   Once the service is added to the gateway, **Commit Policy** appears in red on the page listing currently registered gateways, prompting you to update the Policy Manager with new information.

9. To update changed policies on the gateway, click **Commit**.

After you have added a service to a gateway, you can edit the policy for the service. See Chapter 5, "Managing Oracle Web Services Manager Policies" for more information.

Using the Web Service ID in Client Requests

When you register a Web service, Oracle WSM assigns a service ID for the Web service. Provide this service ID to any **client** that will be making requests to this Web service so that the service ID can be included in the message request. The service ID may be specified through one of the following options:

- For **Java Message Service (JMS)** clients, use a JMS user header with the name `serviceId`, and set the namespace to `http://schemas.confluentsw.com`. Also, you must set the `ResponseQueue` name in the JMS ReplyTo header so that the client can pick up the response message.
- For HTTP clients, specify the `serviceID` as part of the URL.
- For MQ clients, specify the local-tag `serviceID` in the **SOAP header**, and set the namespace to `http://schemas.confluentsw.com`. Also, you must set the `ResponseQueue` name in the MQ `ReplyTo` header so that the client can pick up the response message.

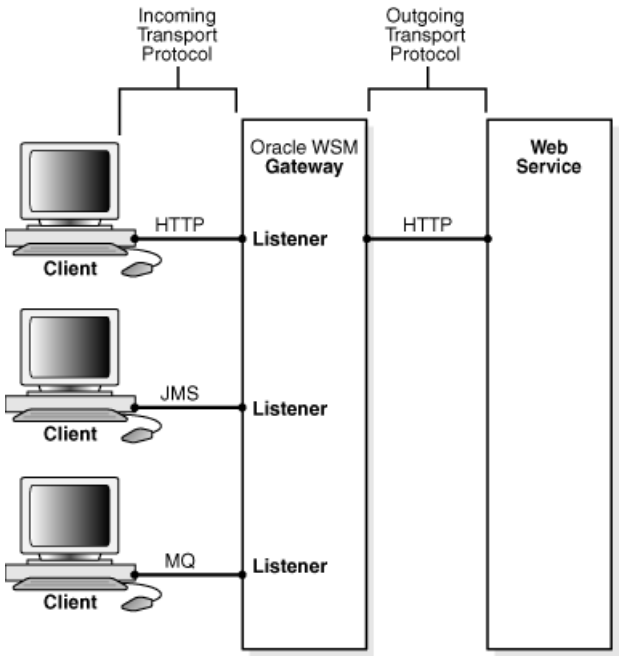
Alternatively, you can use the Web service name instead of the service ID to identify the Web service in client requests. For example, both of the following URLs could be used to identify the TimeService Web service whose service ID is SID0003006:

- `http://jdoe.us.oracle.com:3115/gateway/services/SID0003006?wsdl`
- `http://jdoe.us.oracle.com:3115/gateway/services/TimeService?wsdl`

Oracle WSM Transport Protocols

There are two places where transport protocols are used as shown in Figure 2–1. The first is between the client making the request and the Oracle WSM Gateway listening for requests. The protocol on which the gateway listens for client requests is called the incoming transport protocol. The second is between the Oracle WSM Gateway and the Web service. This is the transport protocol on which the Web service is invoked. Saying this another way, this is the transport protocol on which the gateway sends requests to the Web service. This is called the outgoing transport protocol.

Figure 2–1 Transport Protocols in Oracle WSM



The incoming and outgoing transport protocols are independent of one another. The incoming transport protocol and the outgoing transport protocols can be different, and if they are, then the gateway takes the client request and translates it to the outgoing

transport protocol before passing the request to the Web service. For example, in Figure 2–1, a request coming in on JMS is translated to HTTP before the request is passed to the Web service. However, the request coming in on HTTP does not require translation because it matches the outgoing transport protocol.

The incoming and outgoing transport protocols are configured separately and through different properties files. The outgoing transport protocol is configured through Web Services Manager Control. This is described in the sections that follow.

Configuring the Incoming Transport Protocol

When the Oracle WSM Gateway is created and registered, it is configured to accept requests on HTTP by default; this cannot be modified. Therefore, the gateway always accepts requests on HTTP. In addition the gateway can be configured to listen for requests on two other supported transport protocols: JMS and MQ. You must specifically configure your gateway to listen for requests for JMS and MQ if you expect to receive client requests on these protocols.

The Oracle WSM Gateway can be configured to listen to requests on JMS or MQ by manually configuring the *following file:*

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

Set the properties for one or both protocols, JMS and MQ, in this file. Then redeploy the gateway application for the settings to take effect. All services registered with the gateway will be available to requests coming in on the configured protocols as well as requests coming in on HTTP. See *Oracle Web Services Manager Deployment Guide* for more information on how to configure the incoming transport protocol.

Configuring the Outgoing Transport Protocol

You configure the outgoing transport protocol, that is, the transport protocol on which the Web service is invoked, at the time you register a service with the gateway. You specify one of the following protocols: HTTP, HTTPS, JMS, MQ, or a custom protocol.

If you specify a custom protocol, you must first create a custom policy **step** that implements the protocol for communicating with the Web service. Then, when you register the service, you specify this policy step in the Custom Protocol Step Template Id property. See *Oracle Web Services Manager Extensibility Guide* for more information on creating custom policy steps.

When more than one incoming transport protocol is configured for a gateway, then all registered Web services are exposed to all protocols. In this situation, you cannot configure your Web service to accept requests coming in only on a particular protocol. Because gateways always accept requests on HTTP, Web services are always exposed to requests on HTTP. In the example shown in Figure 2–1, the gateway is also configured to listen for client requests on JMS and MQ. Therefore, in Figure 2–1 the Web service must accept all requests coming in on the JMS and MQ protocols, in addition to HTTP.

See "Registering Web Services to an Oracle Web Services Manager Gateway" on page 2-1 for more information on configuring the outgoing transport protocol.

HTTP and HTTPS

When you register the Web service to the gateway and you specify the protocol used to invoke the Web service (outgoing transport protocol) as HTTP(S), you set the following properties for the protocol (Table 2–1). (See "To add a Web service to a gateway" on page 2-2 to see the procedure where these properties are specified.)

Table 2–1 HTTP(S) Properties

Property	Description
Enabled	Enables this step, if set to true.
URL	Endpoint URL of the Web service.
ReplyTimeout	Length of time, in milliseconds, that the gateway attempts to connect to the Web service before it times out.
IsSoapService	Specifies that the Web service supports SOAP messages, if set to true. If set to false, then the Web services accepts only **XML** messages.
ForwardCredentials	Specifies that HTTP Basic Credentials are forwarded, if set to true.
Failover URLs	Comma-delimited list of URLs to which the gateway tries to connect if the primary URL, specified with the URL property, cannot be reached in the number of attempts specified with the Attempts property.
Attempts	Number of times the gateway tries to invoke a service at a URL.
RetryInterval	Length of time, in seconds, between retry attempts.
KeepAlive	Specifies that the HTTP connection is reused, if set to true.

JMS Messenger

The JMS request handler forwards the request to the designated Web service based on the service ID that is assigned to the JMS service when you register the service. The JMS service ID is a string that you specify in the JMS message header. The Oracle WSM Gateway reads the JMS properties specified when the service was registered and forwards the message to the specified server. Table 2–2 lists the JMS Messenger properties that get specified during registration.

When you register the Web service to the gateway and you specify the protocol to be used to invoke the Web service (outgoing transport protocol) as JMS Messenger, you set the following properties for the protocol to route service invocations to the intended JMS server. (Table 2–1). (See "To add a Web service to a gateway" on page 2-2 to see the procedure where these properties are specified.)

Table 2–2 JMS Messenger Properties

Property	Description
Enabled	Enables this step, if set to true.
OneWay	Specifies that the gateway sends a synchronous request to JMS and terminates the connection, if set to true. It does not care if there is a response.
ReplyTimeout	Length of time, in milliseconds, that the gateway attempts to connect to the Web service before it times out.
IsSoapService	Specifies that the Web service supports SOAP messages, if set to true. If set to false, then the Web services accepts only **XML** messages.
JmsQueueName	Name of the queue on which the Web service listens.
JmsReplyToQueueName	Name of the queue to which the Web Service sends the response. If the value is specified as Dynamic, the response is sent to a temporary queue.
JmsIsPersistent	Specifies that the message persists across JMS server restarts, if set to true.

Table 2–2 (Cont.) JMS Messenger Properties

Property	Description
JmsCorrEnabled	Specifies that the request and response are correlated, if set to true.
UseJndi	Specifies that Java Naming and Directory Interface (JNDI) lookups are performed, if set to true.
JmsUrl	URL used to connect to the JMS server.
JmsConnectionFactory	`ConnectionFactory` class name or JNDI name for opening JMS server connections.
JmsUsername	User name to be used for connections to the JMS server.
JmsPassword	Password to be used for connections to the JMS server.
OwsmServiceID	For internal use only. Do not edit this value.
JndiProviderUrl	JNDI provider URL for getting the JNDI context.
JndiProviderFactory	JNDI provider factory class used to get the JNDI context. You must specify the full path to the Java class. For example, `com.tibco.tibjms.TibjmsQueueConnectionFactory`.
JndiUrlPackagePrefix	Prefix used to initialize the JNDI context.
JndiUsername	User name used to do the JNDI lookup.
JndiPassword	Password used to do the JNDI lookup.
SslEnabled	Specifies that the connection to the JMS server is through SSL, if set to true.
SslVendor	Vendor of SSL certificates. Valid values are: j2se (Java 2 Platform, Standard Edition) and entrust6 (Entrust for TIBCO JMS).
SslHostname	Host name of the JMS server SSL certificate.
SslClientIdentity	Full path to the client certificate.
SslPassword	Password to decrypt the identity file.
SslTrustCerts	Trust certificate of the JMS server or the path to the certificate file.

MQ Series

When you register the Web service to the gateway and you specify the protocol used to invoke the Web service (outgoing transport protocol) as MQ Series, you set the following properties for the protocol to route service invocations to the appropriate MQ server (Table 2–1). (See "To add a Web service to a gateway" on page 2-2 to see the procedure where these properties are specified.)

Table 2–3 MQ Series Properties

Property	Description
Enabled	Enables this step, if set to true.
MqServerHost	Host name of the MQ server.
MqServerPort	Port on which the MQ server listens for Web service requests.
MqUserId	User ID to log in to the MQ server.
MqPassword	Password to log in to the MQ server.
ChannelName	Name of the channel to which to connect on the target queue manager.

Table 2–3 (Cont.) MQ Series Properties

Property	Description
QueueManager	Name of the queue manager on which the queue is defined.
ReplyTimeout	Length of time, in milliseconds, that the gateway attempts to connect to the Web service before it times out.
IsSoapService	Specifies that the Web service supports SOAP messages, if set to true. If set to false, then the Web services accepts only XML messages.
MqReplyToQueueName	Name of the queue to which the Web service sends the response.
MqQueueName	Name of the queue on which the Web Service listens for requests.
MqCorrEnabled	Specifies that the request and response are correlated, if set to true.
MqCharacterSet	Character set used to encode messages that are sent to the MQ server.

Changing the Protocol for a Web Service

If you change the transport protocol for a Web service, you must first deactivate the service, then register it with the new protocol. See "Deactivating a Web Service" on page 9-2 for more information.

Using HTTPS to Secure Connections

Oracle recommends that HTTPS be used to secure transmission of passwords between all communication points in the Oracle WSM environment. See "HTTP and HTTPS" on page 2-4 for specific information on how to configure HTTPS for your Web services.

JMS Server Failover

Oracle WSM supports JMS server failover for TIBCO only. There are two failover mechanisms:

- **Heartbeat Failover** – The primary JMS server sends heartbeat messages to the backup server to verify that the backup server is operational. If a network failure prevents the servers from communicating with each other, the backup server detects the interruption in the stream of heartbeat messages, and it takes on the role of the primary server. The backup server uses the same domain name system (**DNS**) and Internet Protocol (**IP**) address as the primary server. JMS clients are unaware that there is a backup server, because both servers share the same URL. This type of failover requires hardware support, that is, network cards. No changes are required to the gateway to take advantage of a heartbeat failover. When you configure the transport protocol for the service, you specify a single URL for the `JmsUrl` property.
- **Connection Failover** – If the JMS client fails to connect to a primary server, then it tries to connect to one or more backup servers. When the Web service is registered to the gateway and the JMS protocol is configured, multiple URLs are specified for the JMS server. A comma-delimited list of URLs are specified for the `JmsUrl` property. The gateway attempts to connect to the URLs in the order listed. If a connection to the first URL fails, then the gateway tries to connect using the next URL in the list, and continues through the list until it connects or all URLs have been tried. If a connection is not established after all URLs have been attempted,

then the connection fails. See "Adding a Service to an Oracle Web Services Manager Gateway" on page 2-2 and "JMS Messenger" on page 2-5 for more information.

3

Discovering Web Services

Oracle Web Services Manager (Oracle WSM) provides support for discovering and registering Web services that are published in Universal Description, Discovery, and Integration (**UDDI**) registries and in WS-Inspection (WSIL) documents. When you register Web services to a gateway, you can get information on Web services by specifying any of the following:

- URL to a UDDI registry
- URL to a WSIL document
- File location of a WSIL document

Oracle WSM reveals the services specified, the desired services can then be selected and registered to the gateway.

This chapter includes the following sections:

About Web Services Inspection Language

A key feature of the Web services model is the ability to make Web services widely available and discoverable. UDDI is one approach to publishing and discovery of Web services that centralizes information about businesses and their services in registries. Another emerging alternative standard is the Web Services Inspection Language (WSIL) specification.

> **Note:** WS-Inspection and WSIL (Web Services Inspection Language) are used interchangeably in this section.

WSIL defines an Extensible Markup Language (XML) format for referencing Web **service description**s. These references are contained in a WSIL document, and refer to Web service descriptions (for example, WSDL files) and to other aggregations of Web services (for example, another WSIL document or a UDDI registry).

WSIL documents are typically distributed by the Web **service provider**. These documents describe how to inspect the provider's Web site for available Web services. Therefore, the WSIL standard also defines rules for how WSIL documents should be made available to consumers of Web services.

The WSIL model decentralizes Web service discovery. In contrast to UDDI registries, which centralize information on multiple business entities and services, WSIL makes it possible to provide Web **service description** information from any location. Unlike

UDDI, WSIL is not concerned about business entity information, and does not require a specific service description format. It assumes that you know who the service provider is and relies on other standards for Web service description, such as WSDL.

Discovering Web Services

Another way to register Web services to the gateway is to import the services directly from a UDDI registry for published available Web services or from a WSIL document. This is described in the following sections of this chapter:

- Importing Web Services from a UDDI Registry on page 3-2
- Importing Web Services from a WSIL File on page 3-3
- Importing Web Services from a WSIL URL on page 3-5

Note: If you import a Web service by specifying an external URL, you must set your HTTP proxy settings. See "Error When Importing WSIL" on page 10-2 for more information.

Importing Web Services from a UDDI Registry

The following procedure describes how to import Web services from a UDDI registry.

To import a Web service from a UDDI registry

1. In the left navigation pane, select **Policy Management,** then click **Register Services**.

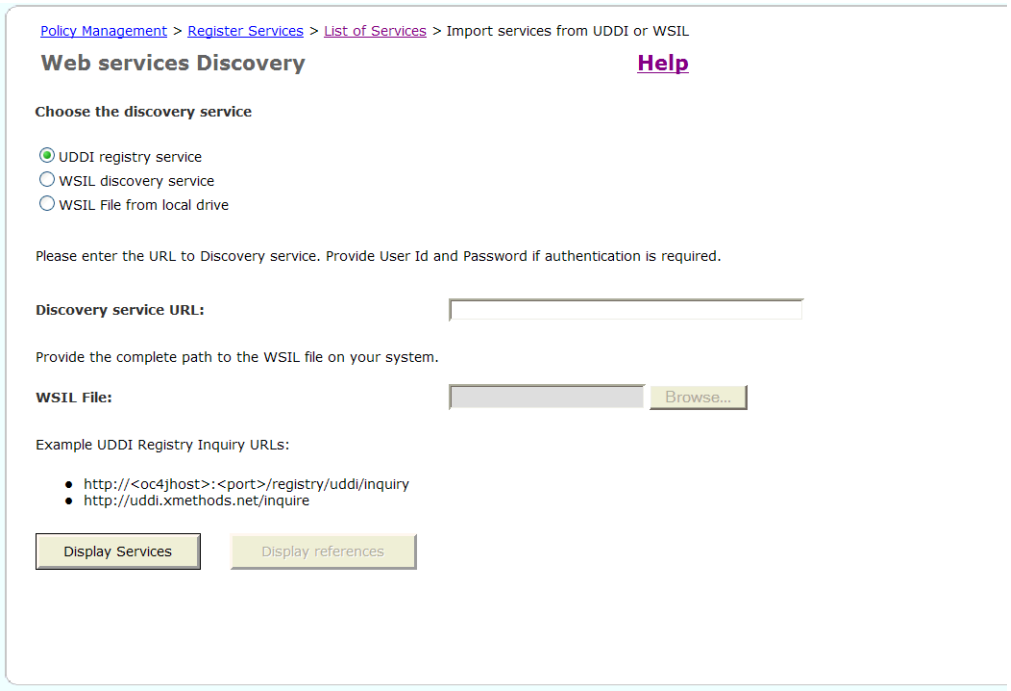
   The Web Services Manager Control displays the list of registered gateways.

2. Find the gateway to which you want to register a service, and click **Services**.

   The Web Services Manager Control lists the gateway's currently registered services.

3. Click **Import Services.**

Figure 3–1 Web Services Discovery Page with UDDI Registry Service Selected



4. In the Web services Discovery page, select **UDDI registry service.**
5. Enter the URL of the service you want to **query** in the Discovery service URL field.

> **Note:** Before you can use the example UDDI provided, `http://<oc4jhost>:<port>/registry/uddi/inquiry`, you must install the Oracle Registry.

6. Click **Display Services** to see a list of the services available from the UDDI registry.
7. Click **Import Services**.
8. Select the check boxes for the Web services you want to import, and then click **Import**.

   The maximum number of services you can import at one time is 50. The process of importing services into Oracle WSM can take several minutes. When the selected services have been imported, an *Import completed* message is displayed.
9. Click **OK**.

   The list of services includes the Web services that you imported from the UDDI registry. Oracle WSM assigns default management policies for the new services. **Commit Policy** appears in red to prompt you to accept the policy updates.
10. Click **Commit**, then click **OK.**

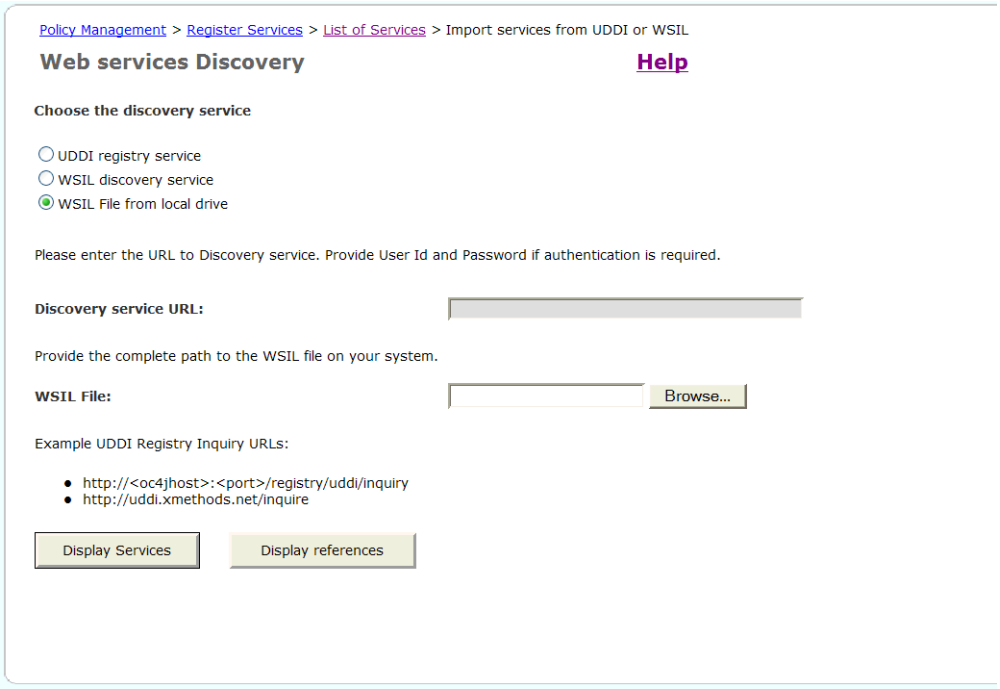
Importing Web Services from a WSIL File

You can either specify a URL or a system path to a WSIL file.

To import a Web service from a WSIL file

1. In the left navigation pane, select **Policy Management,** then click **Register Services**.
2. Click **Services** for the gateway to which you want to register a service.
3. Click **Import Services.**
4. In the Web services Discovery page, select **WSIL File from local drive.**

Figure 3–2 Web Services Discovery Page with WSIL File from Local Drive Selected



5. Enter the path to the WSIL document in the WSIL File field, or click **Browse** to locate the file. You can specify any WSIL document on any system that is accessible from Web Services Manager Control.

   The WSIL document may contain service descriptions or links to other service pointers:

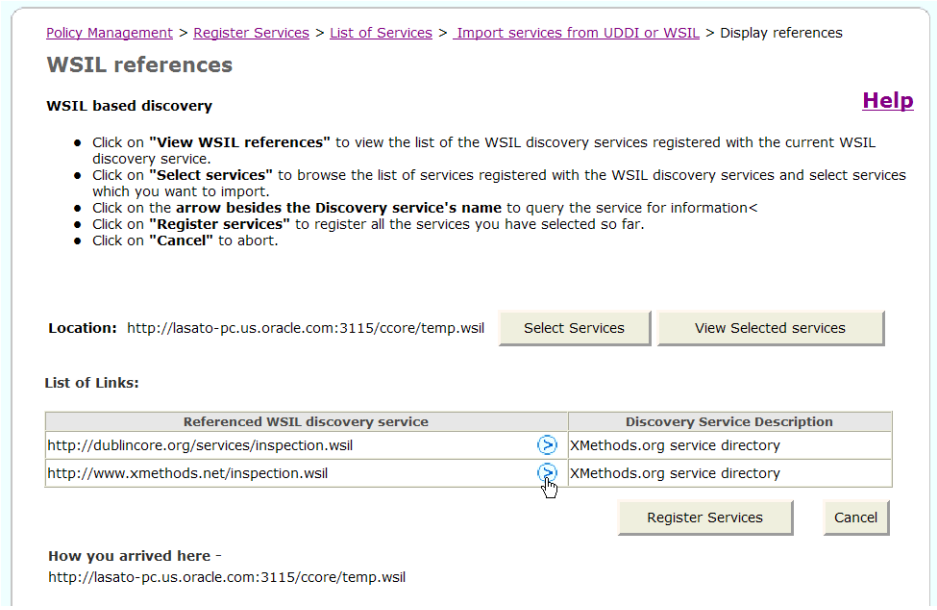
   - To import from a list of the services defined in the WSIL document, go to step 6.
   - To see a list of links to other aggregations of service pointers such as WSIL files, go to step 8.

6. Click **Display Services** to display a list of services.
7. Select the services you want to import from the List of Services, and click **Select.**

   Continue to step 11.

8. Click **Display references** to display a list of pointers to additional WSIL documents.
9. Click the right arrow to follow the link to the service pointer, and view the services that have been aggregated in that service pointer.

Figure 3–3 WSIL References Page Showing WSIL Discovery Services



A list of the services is displayed.

10. Select the services you want to import and click **Select.**
11. Click **Register Services**.

    The process of importing services into Oracle WSM can take several minutes. When the selected services have been imported, an *Import completed* message is displayed.
12. Click **OK**.

    The list of services includes the Web services that you imported from the UDDI registry. Oracle WSM assigns default management policies for the new services. **Commit Policy** appears in red to prompt you to accept the policy updates.
13. Click **Commit**, then click **OK.**

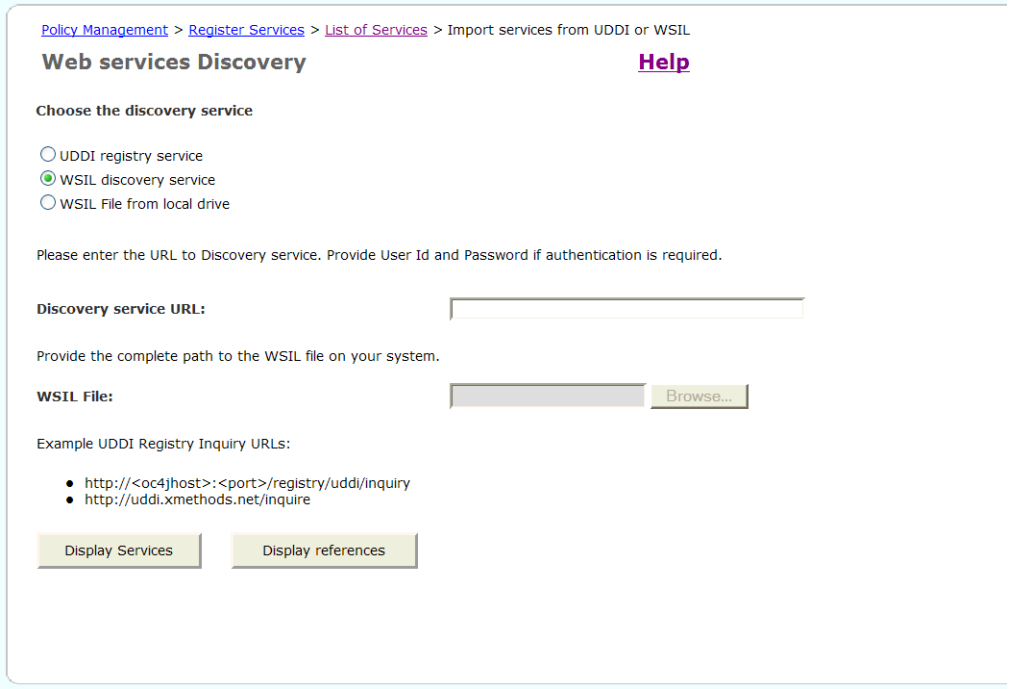
Importing Web Services from a WSIL URL

The following procedure describes how to import Web services from a WSIL URL.

To import a Web service from a WSIL URL

1. In the left navigation pane, select **Policy Management,** then click **Register Services**.
2. Click **Services** for the gateway to which you want to register a service.
3. Click **Import Services.**
4. In the Web services Discovery page, select **WSIL discovery service.**

Figure 3–4 Web Services Discovery Page with WSIL Discovery Service Selected



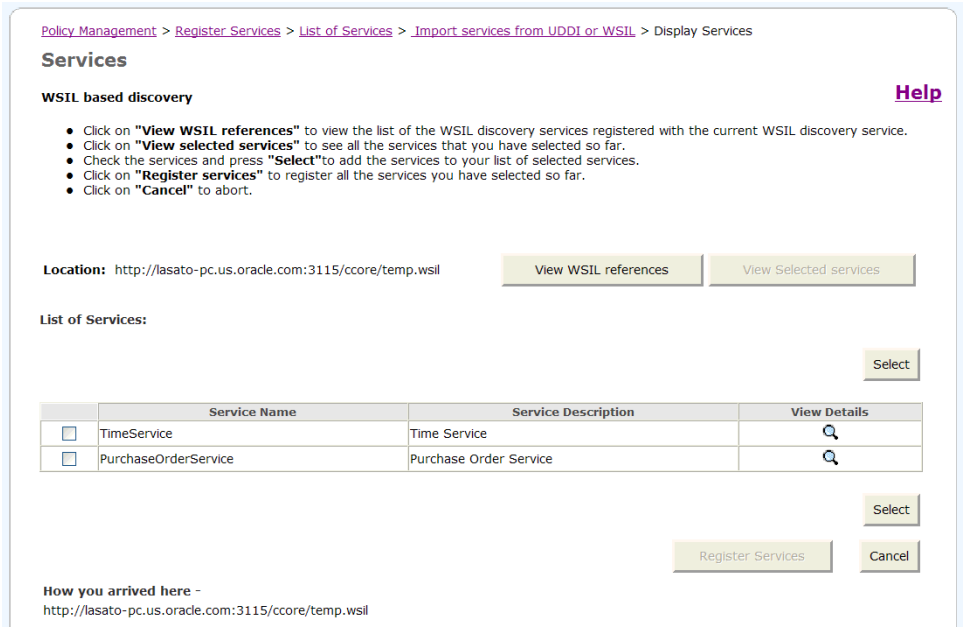
5. Enter the path to the WSIL document in the **WSIL File** field.

   The WSIL document may contain service descriptions or links to other service pointers:

   - To import from a list of the services defined in the WSIL document, go to step 6.
   - To see a list of links to other aggregations of service pointers such as WSIL files, go to step 8.

6. Click **Display Services** to display a list of services.

Figure 3–5 Display Services Page for WSIL-Based Discovery



7. Select the services you want to import from the List of Services, and click **Select.**

   Continue to step 11.

8. Click **Display references** to display a list of pointers to additional WSIL documents.

9. Click the right arrow of the reference to display a list of the services.

10. Select the services you want to import and click **Select.**

11. Click **Register Services**.

    The process of importing services into Oracle WSM can take several minutes. When the selected services have been imported, an *Import completed* message is displayed.

12. Click **OK**.

    The list of services includes the Web services that you imported from the UDDI registry. Oracle WSM assigns default management policies for the new services. **Commit Policy** appears in red to prompt you to accept the policy updates.

13. Click **Commit**, then click **OK.**

4

Content Routing with Oracle Web Services Manager Gateways

This chapter discusses how to route messages to a gateway based on their content. This chapter includes the following sections:

Using Oracle Web Services Manager (Oracle WSM) Gateway content routing rules, you can customize message routing to send messages to different Web service providers based on the content of the actual messages. The dispatching logic of Oracle WSM Gateway routes incoming messages to different pipelines; and, therefore, different service end points, based on the destination URL in the incoming SOAP message. When the incoming message is not a SOAP message, it is normalized (that is, wrapped) to SOAP, so that the dispatching logic remains the same.

The client needs to know only the client access URL through which it sends its message to the gateway. The gateway then routes the message to the Web service URL. The client does not need to know the URL of the Web service. And, in many cases, you may want to hide this URL from the client for the following reasons:

- A gateway may be exposed to business partners, but the Web services are internal and are not published externally.
- You may want to designate the service end point based on content and, therefore, you do not want to expose Web service endpoint URLs to the client.

With content routing, a client sends a message to the gateway without any specific forwarding service address. The routing is based on the XML content in the **SOAP envelope** (header or body) or in a SOAP attachment, and on designated content routing rules. Content routing can be applied to any incoming SOAP message, over any transport protocol, and can be routed to any service registered with the gateway.

When a message arrives at the gateway with a URL prefix that contains */fs*, for example, *http://host:port/gateway/fs/crouter*, the gateway forwards the message to a specific Web service based on the following:

- The content of the message
- The content routing rules configured at the gateway

This service can be invoked with any of the available transports, for example, HTTP(S), JMS, and MQ, using the following methods:

- For HTTP: Send messages to the URL *http://host:port/gateway/fs/crouter.*
- For JMS: Set the services ID in the JMS header or SOAP header to *fs:crouter.*
- For MQ: Set the service ID in the SOAP header to *fs:crouter*.

About Gateway Content Routing Rules

You configure content-based rules to designate the destination of messages sent to a gateway. The gateway evaluates the message based on these rules.

The following is a simple real-world application use case:

A major insurance company has two Web services that process incoming insurance requests for quotes (RFQs):

- Web Service 1 processes RFQs from homeowners.
- Web Service 2 processes RFQs from automobile owners.

The insurance company does not want to reveal the URLs for these two services directly to client applications. Instead, it wants to show a single URL to the client, then have the gateway route the message to either Web Service 1 or Web Service 2 based on the *clientType* field in the RFQ.

A system administrator would create the following rules for the gateway:

- Rule 1: IF /soap:Body/RFQ/clientType = HOMEOWNER THEN ROUTE to service1
- Rule 2: IF /soap:Body/RFQ/clientType = AUTOOWNER THEN ROUTE to service2

When a message arrives at the gateway, the gateway evaluates each rule, in order, and routes the message according to the first rule that matches.

Content routing rules are based on the creation of XPath1 expressions with simple condition matching. A rule consists of a condition that returns either a true or a false value and an action. More complex conditions for the rule can be created by specifying multiple simple conditions that are connected with the Boolean AND expression:

Table 4–1 Rule Creation Elements

Element	Description
Rule	A rule consists of a condition and an action. A condition is a set of one or more simple conditions that, when evaluated, return either a true or false value. The action specifies a single routing operation that is performed if the associated condition is true.
Condition	A condition consists of one or more simple conditions that are connected using the Boolean AND expression to return either a true or a false value. Simple conditions can be defined using the following operators: EQUALS – Value of an XML element equals X (string match only) NOT_EQUALS – String value on an XML element does not match X EXISTS – XML element exists in SOAP header or body or attachment NOT_EXISTS – XML element does not exist in SOAP header or body or attachment

Table 4–1 (Cont.) Rule Creation Elements

Element	Description
Action	If the associated condition is true, the action routes or forwards the message to the specified Web service.

Using Web Services Manager Control, you can do the following:

- Add a new content routing rule by specifying a new condition (variable-XPath, conditional operator, value) or specifying the name space prefix, if necessary, for defining the variable-XPath.
- List and view all the content routing rules.
- Delete a content routing rule.
- Change the priority order in which content routing rules are evaluated.
- Specify the address for target Web services.

Content routing rules use the following syntax:

Conditions: IF <*variable*> <EQUALS/EXISTS> <*value*>

Namespaces: WHERE <*prefix*> = <*namespace_url*>

Action: THEN Route To <*serviceId*>

Namespaces

Application namespaces vary from application to application and are entirely dependent on the design guidelines of the application.

- SOAP-ENV is a namespace that points to `http://schemas.xmlsoap.org/soap/envelope/`

Sample SOAP Message

The following code sample show a typical SOAP message:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <r:RFQ xmlns:r="http://businessdocs.com/RFQ">
         <clientType>HOMEOWNER</clientType>
         <clientID>12345</clientID>
      </r:RFQ>
   </soap:Body>
</soap:Envelope>
```

Sample Content Routing Rules

The following provides a description of two sample content routing rules that could be defined for the previous sample SOAP message:

1. If `/soap:Envelope/soap:Body/r:RFQ/clientType` equals `HOMEOWNER` where the SOAP namespace is `http://schemas.xmlsoap.org/soap/envelope/` and the `r` namespace is `http://businessdocs.com/RFQ`, then route to `Service1`.
2. If `/soap:Envelope/soap:Body/r:RFQ/clientType` equals `AUTOOWNER` where the SOAP namespace is

`http://schemas.xmlsoap.org/soap/envelope/` and the `r` namespace is `http://businessdocs.com/RFQ,` then route to `Service2`.

Accessing the Web Service Using Content Routing

This section describes how to access a Web service using content routing.

To access the Web service using content routing

1. From the navigation pane of Web Services Manager Control, select **Tools,** then click **Test Page.**
2. Enter the URL for a Web service that is registered to the gateway, and click **Submit Query.**

   The Test Web Service page refreshes and displays the endpoint URL as well as other parameters that you can set.
3. Replace the endpoint URL with the following URL:

   `http://`*`host_name:port`*`/gateway/fs/crouter`
4. Click **Invoke.**

Creating Rules for Oracle WSM Content Routing

The procedure that follows creates content routing rules for the sample SOAP message:

If `/soap:Envelope/soap:Body/r:RFQ/clientType` equals `HOMEOWNER` where the SOAP namespace is `http://schemas.xmlsoap.org/soap/envelope/` and the `r` namespace is `http://businessdocs.com/RFQ,` then route to `Service1.`

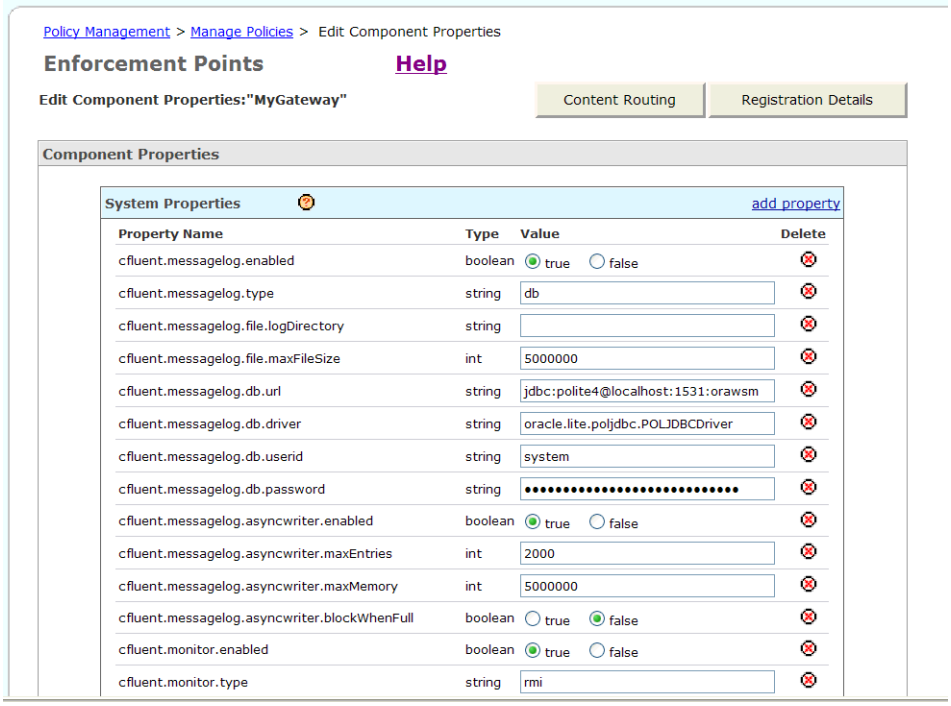
To create rules for Oracle WSM content routing

1. Start Web Services Manager Control, logged in as Component Administrator for the gateway for which you want to define content routing rules.
2. In the navigation pane, select **Policy Management,** then select **Manage Policies.**

   The Web Services Manager Control displays a list of currently registered Oracle WSM components, including gateways.
3. Click **Edit** for the gateway for which you want to add content routing.

   Web Services Manager Control displays the Edit Component Properties page, showing a list of the properties currently assigned for the gateway component.

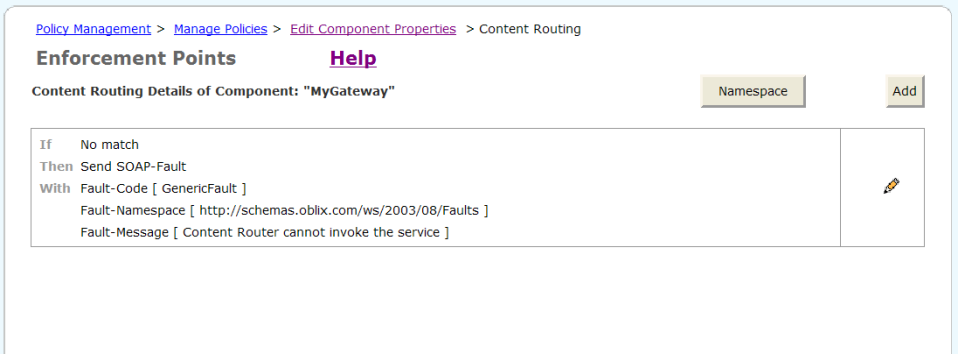
Figure 4–1 Edit Component Properties Page



4. Click **Content Routing**.

   Web Services Manager Control displays a list of currently defined content routing rules for the gateway, if any are defined.

Figure 4–2 Content Routing Details Page



   Oracle WSM provides a default content routing rule based on the condition *No match*, which is invoked if no other content routing condition matches.

5. Click **Add**.

   Web Services Manager Control displays an empty Add/Edit Rule page where you specify the conditions and actions for a new content routing rule. By default, the SOAP Envelope option is selected.

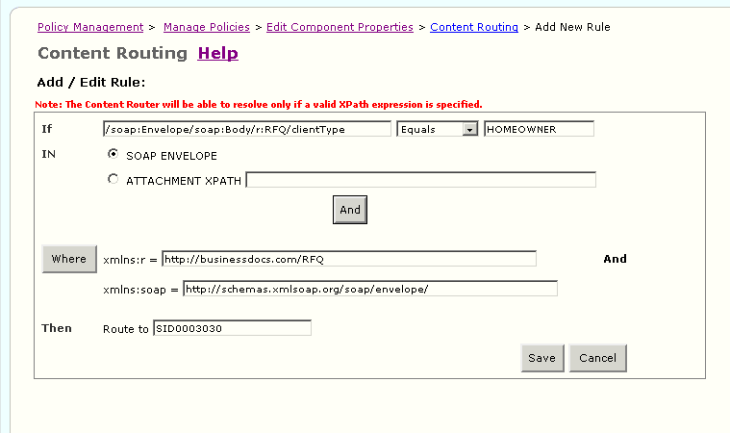
6. Enter the IF condition in the SOAP message in the **IF** field, and enter the Web service to which to route the message in the **THEN** field.

Note: To route the message to a specific service location, you can specify its service name (specified during service registration) or the service ID number (SID).

7. Click **WHERE** and specify the namespaces for any XPath variables included in the IF conditions of the rule (Figure 4–3).

   In the insurance example, the first content routing rule is: If `/soap:Envelope/soap:Body/r:RFQ/clientType` equals `HOMEOWNER` where the SOAP namespace is `http://schemas.xmlsoap.org/soap/envelope` and the `r` namespace is `http://businessdocs.com/RFQ` then route to `Service1`.

Figure 4–3 Content Routing Page



8. When you have finished your entries, click **Save.**

   Web Services Manager Control displays the content routing rules, including any new rules you added.

   The order in which rules are listed reflects the order in which content routing rules are applied to a message sent to the gateway. Since the incoming message is routed according to the first rule that Oracle WSM Gateway verifies as true, you may want to change the order of the listed rules to match whatever selection criteria is best suited for your environment.

9. Click on the up or down arrow to change the order of the rules.

When you click **Namespace**, a window opens that displays all **XML namespaces** used across all the rules defined for this gateway.

Figure 4–4 Namespace Mappings for Content Routing Rules

A prefix can be mapped to only one namespace across all content routing rules for the gateway. In Figure 4–4, there are two prefixes, *r* and *soap*. You cannot, for example, have the same prefix, both named *soap*, mapped to different namespaces.

Creating Rules Using Attachment XPath Content

In addition to defining content routing rules based on the XML content of SOAP messages, you can also define rules that are based on the XPath content of SOAP attachments.

To create rules using attachment XPath content

1. Start Web Services Manager Control, logged in as Component Administrator for the gateway for which you want to define content routing rules.
2. In the navigation pane, click **Policy Management,** then click **Manage Policies.**

   Web Services Manager Control displays a list of currently registered Oracle WSM components, including gateways.
3. Click **Edit** for the gateway for which you want to add content routing.

   Web Services Manager Control displays the Edit Component Properties page, showing a list of the properties currently assigned for the gateway component.
4. Click **Content Routing**.

   Web Services Manager Control displays a list of currently defined content routing rules for the gateway, if any are defined.
5. Click **Add**, then select the **ATTACHMENT** XPath option

   The Web Services Manager Control displays the Add/Edit Rule page.
6. Specify the associated XPath variable, contained in the SOAP message, that identifies attachments to the message. For example:

   ```
   /soap:Envelope/soap:Body/source/@href
   ```
7. Enter one or more rule conditions to match the attachment content. In the **IF** field, specify XPath variables found in the attachment. For example:

   ```
   IF /policy/model EQUALS lexus
   ```
8. Click **WHERE** and specify the namespaces for any XPath variables included in the IF conditions of the rule. For example:

   ```
   http://schemas.xmlsoap.org/envelope
   ```

9. Specify the Web service to which to route the message in the **THEN** field.
10. Click **Save.**

    The new attachment XPath rule now shows up in the display of defined gateway content routing rules.

Note: Attachment XPath content routing rules can be defined only for attachments with XML content. These attachments must be referenced inside the SOAP message using the SOAP-with-Attachments specification.

5

Managing Oracle Web Services Manager Policies

This chapter describes the process for creating and editing the policies that enforce security on policy enforcement points, and for attaching those policies to Web services that are registered with the Oracle WSM Gateways and Oracle WSM Agents. This chapter includes the following sections:

Policy Management Overview

Oracle WSM **policies** are sets of operational tasks that are performed at specified policy enforcement points during the processing of Web service requests between a service client and service provider. Each operational task is implemented as a policy step that addresses a specific operation, such as **authentication** and **authorization**, encryption and decryption, security signature, token, or credential verification, and transformation, that is performed on either a Web service request or a response to a Web service request.

Oracle WSM separates each operational policy into a request **pipeline** and a response pipeline.

- **A Request Pipeline** – A set of policy steps that is executed during the processing of a Web service request.
- **A Response Pipeline** – A set of policy steps that is executed during the processing of a response to a Web service request.

> **Note:** The PreRequest and PostResponse pipelines are disabled and will be discarded in future releases of Oracle WSM.

As the administrator, you can assemble policies from the prepackaged policy steps that come with Oracle WSM, or create custom policy steps.You can assign these created policies to the desired Oracle WSM policy enforcement points—either an Oracle WSM gateway or agent. The Oracle WSM Policy Manager lets you enforce policies in a consistent manner across multiple gateways or agents, throughout the system. You can specify operational policies as needed for each Web service or group of services.

Oracle WSM Policy Steps

Oracle WSM comes with a predefined set of configurable policy steps to use in creating your pipelines. This section describes the predefined Oracle WSM policy steps, organized by their intended use. It includes steps for integrating third-party products and technologies, where appropriate.

The predefined policy steps fall into the following general categories:

Security Steps

- Credential Management
- Authentication
- Authorization
- Integrity and Confidentiality
- Federation

Nonsecurity Steps

- Log messages
- Custom fault messages
- Message transformation using XSLT

Refer to Appendix A, "Oracle Web Services Manager Policy Steps" for more information on each of the policy steps.

Supported SOAP Versions

Oracle Web Services Manager supports SOAP version 1.1 only.

Viewing Policy Steps

You can view the available policy steps and their properties for the different policy enforcement points.

To view the available policy steps

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

   A list of registered Oracle WSM gateways and agents is displayed.

2. In the row for a particular gateway or agent, click **Steps.**

   Oracle WSM displays a list of policy steps for the gateway or agent and a brief description of what the step does.

3. To view the properties for a particular policy step, click **Details.**

   The name, data type, and default values for the properties for the step are displayed. Click the question mark (**?**) for help on how to configure the properties.

Credential Management

Message credentials, in this case user name and password combinations, are delivered in various ways; through the HTTP transport headers, SOAP headers, or in the XML body. The first step in building a policy that enforces Web service security, is usually to include the Extract Credentials Step. This step is configured to extract the message credentials that are being delivered through different methods: the HTTP transport headers, SOAP headers, or through XML. Oracle WSM then extracts the user name and password credentials to be used in the next step, which is usually authentication.

> **Note:** When you **deploy** Oracle Access Manager in conjunction with Oracle WSM, do not use the Extract Credentials step in Oracle WSM *request* pipelines. This is because Oracle Access Manager sends user credentials to the managed Web service by means of an obSSOCookie when the user logs in. Therefore, Oracle WSM must not duplicate this action.

The steps in Table 5–1 are used to locate the credentials, and extract, add, or manipulate the credential information so that it is presented in an acceptable form for a common means of authentication.

Table 5–1 Credential Management Steps

Step Name	Description
Extract Credentials	Extracts credentials from HTTP transport layer headers, SOAP headers, or XPath.
Insert Oracle Access Manager Token	Inserts an ObSSOCookie in the SOAP security header.
Insert WSBASIC Credentials	Inserts WS-BASIC credentials.

Refer to Appendix A, "Oracle Web Services Manager Policy Steps" for more information on each of the policy steps.

Authentication

Authentication is used to establish proof of identity. A user's credentials, usually a user name and password combination, are checked against a database or LDAP directory, before granting or denying access. In Oracle WSM, you can check credential information against Oracle WSM's native authentication store, Oracle Access Manager, or against a third-party security database such as CA or an LDAP directory. Authentication may also be presented and verified in the form of an X.509 **certificate**.

Table 5–2 Authentication Steps

Step Name	Description
Active Directory Authenticate	Authenticates client's credentials with Active Directory.
File Authenticate	Verifies the sender’s identity by checking against entries in a file.
LDAP Authenticate	Performs authentication with an LDAP directory.
Oracle Access Manager Authenticate Authorize	Authenticates and authorizes requests against an installed Oracle Access Manager Identity System. (See *Oracle Web Services Manager Deployment Guide* for configuration details.)

Table 5–2 (Cont.) Authentication Steps

Step Name	Description
SiteMinder Authentication	For use with CA eTrust SiteMinder authentication system.
Verify Certificate	Verifies if a certificate path is valid by validating the trusted root and intermediate certificates.

Note: The LDAP Certificate Authenticate policy step is no longer supported. However, similar functionality can be achieved by using the Oracle Access Manager Authenticate Authorize policy step.

Refer to Appendix A, "Oracle Web Services Manager Policy Steps" for more information on each of the policy steps.

Note: To use the SiteMinder Authentication step, you must install the CA eTrust SiteMinder **SDK (software development kit)** and have the files required by the Oracle WSM gateway or agent. See *Oracle Web Services Manager Deployment Guide* for more information.

Authorization

Authorization steps ensure that once a client's principal identity is established, the client has permission to access resources. Authorization may also be described as verification of a role. Oracle WSM steps support authorization through a variety of authorization services: LDAP, Oracle Access Manager, CA eTrust SiteMinder, and file-based authorization.

Table 5–3 Authorization Steps

Step Name	Description
Active Directory Authorize	Authorizes the request by retrieving the client's role from Active Directory and checking against the roles allowed by the service.
File Authorize	Grants or denies access to an authenticated client using a local roles file.
Oracle Access Manager Authenticate Authorize	Authenticates and authorizes requests against an installed Oracle Access Manager Identity System. (See *Oracle Web Services Manager Deployment Guide* for configuration details.)
LDAP Authorize	Authorizes the request by retrieving the client's role from the LDAP store and checking against roles allowed by the service.
SiteMinder Authorize	Authorizes clients using CA eTrust SiteMinder. This step is used after the SiteMinder Authentication step.

Refer to Appendix A, "Oracle Web Services Manager Policy Steps" for more information on each of the policy steps.

Note: To use the SiteMinder Authorize step, you must install the CA eTrust SiteMinder SDK and have the files required by the corresponding Oracle WSM gateway or agent. See the *Oracle Web Services Manager Deployment Guide* for more information.

Integrity and Confidentiality

Oracle WSM enables message confidentiality through message encryption and message integrity through digital signatures. The Web Services Security (WS-Security) specification provides a standard set of SOAP extensions that can be used when building secure Web services. Web Services Security specifications protect message content by encrypting or digitally signing the message body, a header, an attachment, or any number of these elements in any combination. Security tokens can also be created and inserted to verify authentication.

Example of Message Integrity

The following code sample shows the typical structure of a signature included in the Security header in conformance with the WS-Security 1.0 standards. In this example, the body element of the SOAP message is signed. The elements that are particularly significant are indicated in bold.

```
<soap:Envelope ...>
  <soap:Header>
   <wsse:Security ...>
     <wsse:BinarySecurityToken wsu:Id="bst-id" ValueType="...#X509v3"
...>MIICtjCCAnQCBET...
     </wsse:BinarySecurityToken>
     <dsig:Signature ...>
       <dsig:SignedInfo>
         <dsig:CanonicalizationMethod ... />
            <dsig:SignatureMethod ... />
              <dsig:Reference URI="#body-id">
               <dsig:Transforms>
                 <dsig:Transform Algorithm="...xml-exc-c14n#" />
               </dsig:Transforms>
               <dsig:DigestMethod Algorithm="...#sha1" />
               <dsig:DigestValue>205Oyy...=</dsig:DigestValue>
              </dsig:Reference>
       </dsig:SignedInfo>
       <dsig:SignatureValue>Y8yo94k8Xp...</dsig:SignatureValue>
       <dsig:KeyInfo>
         <wsse:SecurityTokenReference ...>
        <wsse:Reference URI="#bst-id" />
         </wsse:SecurityTokenReference>
       </dsig:KeyInfo>
     </dsig:Signature>
   </wsse:Security>
  </soap:Header>
 <soap:Body wsu:Id="body-id" ... >
     ....
 </soap:Body>
</soap:Envelope>
```

Example of Message Confidentiality

The following code sample is an example of the typical structure of encryption elements included in the Security header in conformance with the WS-Security 1.0 standards. In this example, the body element is encrypted. The elements that are particularly significant are indicated in bold.

```
<soap:Envelope ...>
 <soap:Header>
  <wsse:Security ...>
   <wsse:BinarySecurityToken wsu:Id="bst-id" ValueType="...#X509v3"
...>MIICtjCCAnQCBET...
```

```
    </wsse:BinarySecurityToken>
    <xenc:EncryptedKey ...>
      <xenc:EncryptionMethod Algorithm="..." />
       <dsig:KeyInfo ...>
        <wsse:SecurityTokenReference ... >
         <wsse:Reference URI="#bst-id" ValueType="...#X509v3" />
        </wsse:SecurityTokenReference>
       </dsig:KeyInfo>
       <xenc:CipherData>
        <xenc:CipherValue>YV19Qkq79Tub...=</xenc:CipherValue>
       </xenc:CipherData>
      <xenc:ReferenceList>
       <xenc:DataReference URI="#body-id" />
      </xenc:ReferenceList>
    </xenc:EncryptedKey>
  </wsse:Security>
 </soap:Header>
 <soap:Body>
  <xenc:EncryptedData Type="...#Content" Id="body-id" ... >
   <xenc:EncryptionMethod Algorithm="..." />
    <xenc:CipherData>
     <xenc:CipherValue>0h3XxkxLKAROdbd...=</xenc:CipherValue>
    </xenc:CipherData>
   </xenc:EncryptedData>
 </soap:Body>
</soap:Envelope>
```

Oracle WSM provides the steps shown in Table 5–4 for message integrity and confidentiality. The steps are listed in logical order rather than in alphabetical order.

Table 5–4 Integrity and Confidentiality Steps

Step Name	Description
Sign Message	Digitally signs the message.
Verify Signature	Verifies the signature of the XML message that was signed to protect the integrity of the message.
XML Encrypt	Enciphers an XML message.
XML Decrypt	Decrypts the XML message or parts of the message that were encrypted for confidentiality.
Sign Message and Encrypt	Attaches a signature to an XML message and enciphers the message.
Decrypt and Verify Signature	Decrypts the XML message and verifies that the signature is valid.

Refer to Appendix A, "Oracle Web Services Manager Policy Steps" for more information on each of the policy steps.

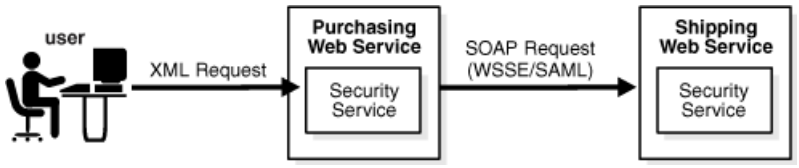
Support for Multiple Keystores

Oracle WSM supports JKS (Java Keystore) and PKCS#12 (Public Key Cryptography Standard #12) keystore file formats for message integrity. To use Oracle wallets, you must use the PKCS#12 keystore file format.

Identity Propagation

When using multiple Web services to complete a single transaction, it is important to propagate the identity of the original requester throughout the transaction. For example, a user may invoke a procurement application to place a purchase order (Figure 5–1). Once the user has filled out the purchase order, the procurement application takes care of formatting the request in a SOAP message, and sends that SOAP message to a provider (for example, an office supplies vendor). When the office supplies vendor has successfully processed the purchase order, it sends a request to a shipping company to deliver the purchased items to the original requester. In this case, the original requester's identity must be propagated throughout the entire transaction for security and audit/compliance purposes. This is achieved by creating a session ticket on behalf of the user, in the form of a **SAML** assertion, including authentication credentials and possibly attributes to be used for authorization.

Figure 5–1 Propagating Requester's Identity Throughout a Transaction

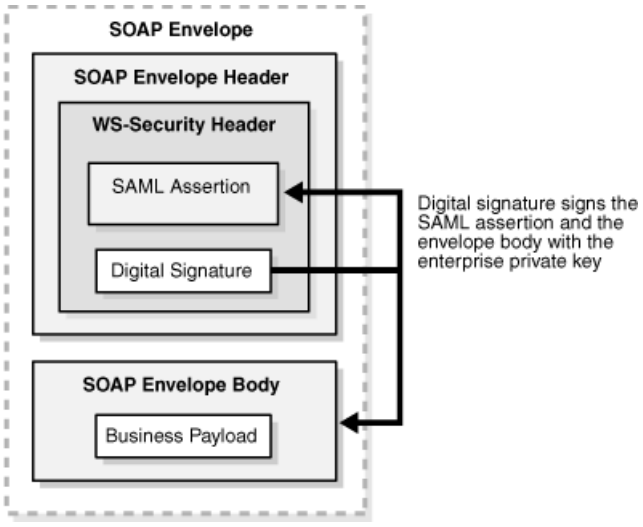


To implement use cases involving identity propagation, Oracle WSM supports the WS-Security SAML Security Token Profile's "Sender-Vouches" scenario. "Sender-Vouches" allows the SOAP message producer to sign the SAML assertion and the envelope body (containing the purchase order, for example), with the original requester's company's private key. In other words, the company "vouches for" the original requester. In this case, the business payload in the SOAP envelope body may be the purchase order used in the example described above.

Figure 5–2 shows how the use case described above can be implemented in a standards-based way:

1. A Security Service authenticates the user request (at the Purchasing Web Service site).
2. The Security Service generates a SAML assertion and inserts it in the WS-Security header as part of the SOAP message.
3. The Security Service signs both the SAML assertion and the body of the message with the Enterprise private key. In this case, the Enterprise is the site that is exposing the Purchasing Web service.
4. The Purchasing Web service processes the request and then posts a SOAP request to the Shipping Web service.
5. At the Shipping Web service, the Security Service authenticates the request. The Security Service checks the signature covering the SAML assertion and the message body, validates that the SAML assertion was issued by a trusted partner, and validates that the user is in the user store.

Figure 5–2 SAML Assertion Within a SOAP Envelope



Oracle WSM can be the Security Service mentioned above, both at the Purchasing site and Shipping site, or at either site. Because WS-Security is an industry standard, Oracle WSM can produce and sign SAML assertions that are consumed by another security service, and it can consume SAML assertions produced by the security service.

Oracle WSM has policy steps that are specifically designed for use with SAML security systems.

Table 5–5 Identity Propagation Steps

Step Name	Description
SAML – Insert WSS 1.0 Sender-Vouches Token	Secures SOAP message by inserting SAML assertions.
SAML – Verify WSS 1.0 Token	Verifies the SAML token according to the Web Services Security SAML Token Profile 1.0 (WSS STP 1.0) standard.

Refer to Appendix A, "Oracle Web Services Manager Policy Steps" for more information on each of the policy steps.

Nonsecurity Policy Steps

Oracle WSM also provides policy step templates for some common processing functions of Web services messaging, including; logging, message persistence, message duplication, and XSLT translation.

Oracle WSM includes a default logging step in its default gateway and agent policy pipelines. As Web services are registered to gateways, or agents are installed into application servers, Oracle WSM creates a default policy pipeline.

Table 5–6 Nonsecurity Steps

Step Name	Description
Handle Generic Fault	Provides custom message in the SOAP fault when errors are encountered.
Log	Logs the current message as received in the policy step.

Table 5–6 (Cont.) Nonsecurity Steps

Step Name	Description
XML Transform	Modifies the incoming XML using an XSLT file.

Refer to Appendix A, "Oracle Web Services Manager Policy Steps" for more information on each of the policy steps.

Adding Policy Steps to Oracle WSM Gateway Web Services

To enable policy enforcement for a Web service at a gateway, you must first register the Web service with the desired gateway. Each registered Web service has its own policy and associated policy steps. When the Web service is registered, Oracle WSM sets up a default policy for the Web service. You can edit the default policy, create a new policy for the service, or create policy pipeline templates, for each Web service. You must have at least Oracle WSM Service Administration permissions to edit policy steps.

To add policy steps to a Web service

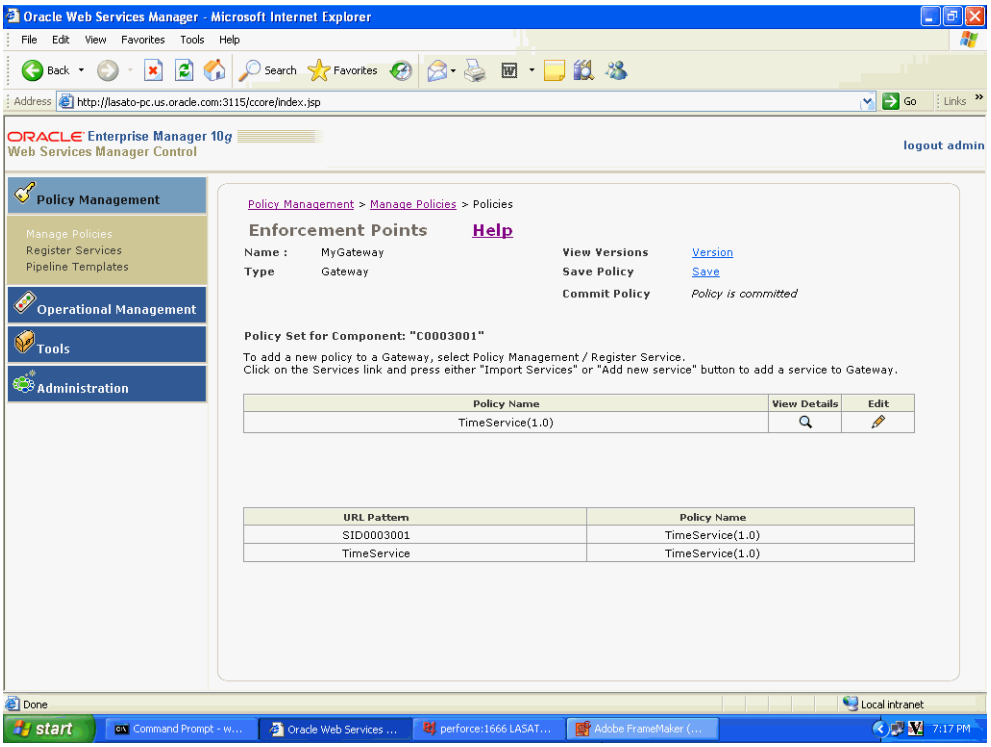
1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Manage Policies.**

   A list of registered Oracle WSM gateways and agents is displayed.

2. In the row for a specific gateway, click **Policies.**

   Oracle WSM displays a list of the current policies registered with the gateway. By default, Oracle WSM creates a policy with the same name and version number as the registered service. In Figure 5–3, the default policy name is TimeService (1.0), which is the same as the name and version of the service that is registered with the gateway.

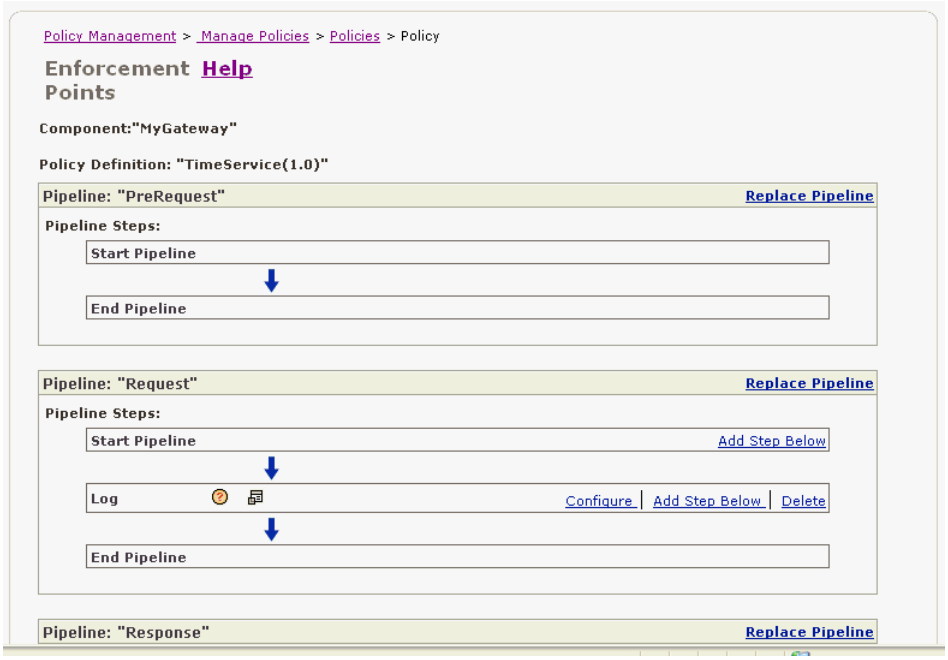
Figure 5–3 Policies for Gateway



3. In the row of the policy you want to update, click **Edit**.

   Web Services Manager Control displays the definition for the selected policy, that is, the steps in the Request and Response policy pipelines (Figure 5–4). Scroll down the page to see the policy pipelines.

Figure 5–4 Policy Definition Page for a Gateway Policy



4. In the pipeline to which you want to add a policy step, click **Add Step Below**.

   The page refreshes and the New Step box appears.

5. Select a step from the Select Step Template list.

   The selected policy step is added to the pipeline.

6. Click **Configure** to configure the properties for the step.

   Note: Click the question mark (**?**) next to the name of the policy step for help on how to configure the properties.

   Click **Environment Properties** to display a list of properties that appear in the form ${*propertyName*} and a description of the properties. You may use items from this list in setting properties in text fields. For example, for the XML Encrypt policy, you could specify the keystore environment property ${acme.keystore} in the **keystore location** field.

7. When you are finished editing the properties, click **OK.**

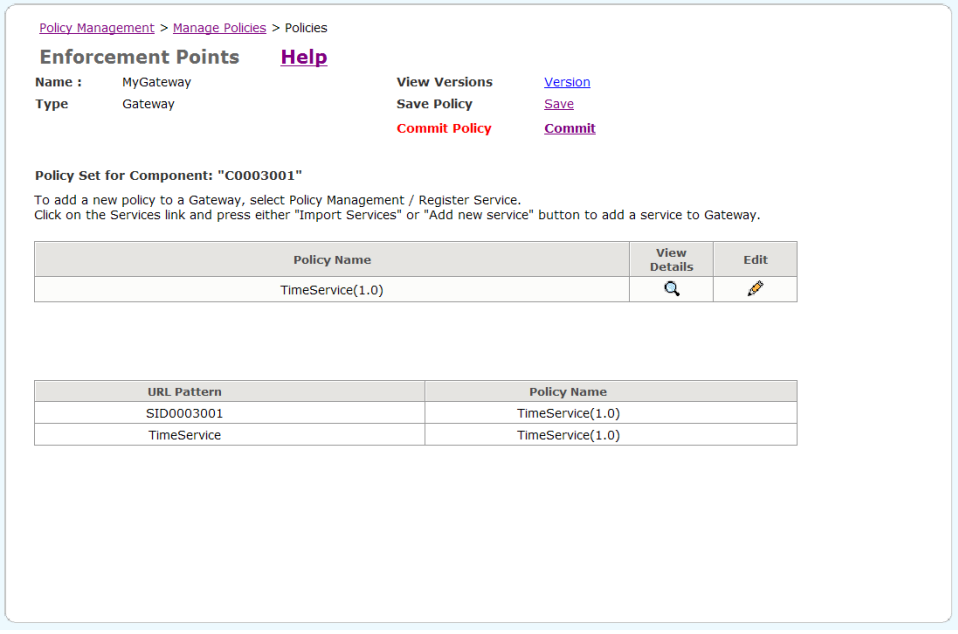
8. Continue to add and configure policy steps.

9. When you are finished, click **Next.**

10. You can either accept the default name of the policy, or enter a different name, then click **Save.**

11. The Commit Policy field appears in red, alerting you to commit your changes (Figure 5–5). Click **Commit** to update the policy for the associated gateway.

Figure 5–5 Committing Changes to the Policy



The *Policy is committed* message is displayed.

Note: You can click **Save** to save the policy to a file. You can then open this file to see the XML representation of the policy. Or you can configure the agent or gateway to run in disconnected mode and refer to this file for the policies, rather than run in connected mode where the **policy enforcement point (PEP)** connects to the Oracle WSM Policy Manager.

Creating Policies for Oracle WSM Agents

Each Oracle WSM agent has its own policy and associated policy steps. Oracle WSM assigns a Default Policy when you first register an agent. This policy, by default, is applied to all services hosted by Oracle Application Server. In most instances, Oracle Application Server hosts one Web service. If you want one policy to be used for all Web services, you simply edit the Default Policy.

Less common is a situation where the Application Server hosts multiple Web services, and you may want a different policy to be applied to each Web service. In this situation, you would create as many policies as necessary, then map the policy to the appropriate Web service. See "Assigning Policies to Web Service URLs for Agents" on page 5-15.

Defining Policies for Oracle WSM Agents

This section describes how to define policies for Oracle WSM Agents.

To define a policy for an Oracle WSM agent

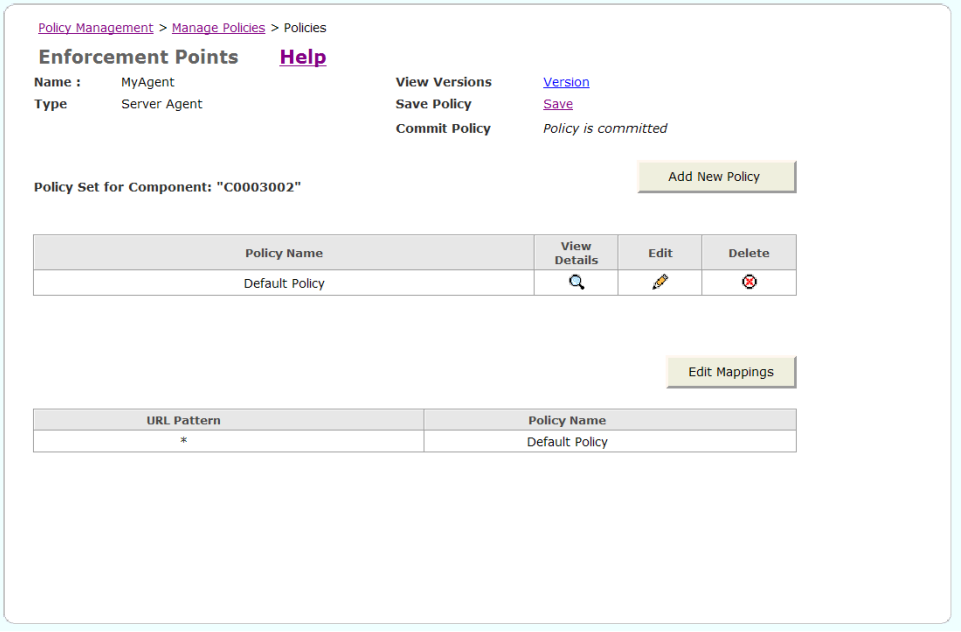
1. In the left navigation pane of Web Services Manager Control, select **Policy Management,** then select **Manage Policies.**

   A list of registered gateways and agents is displayed.

2. In the row for a specific agent, click **Policies**.

   Oracle WSM displays the policies for the selected agent. If no policies have been created for the agent, only the Default Policy appears in the list (Figure 5–6).

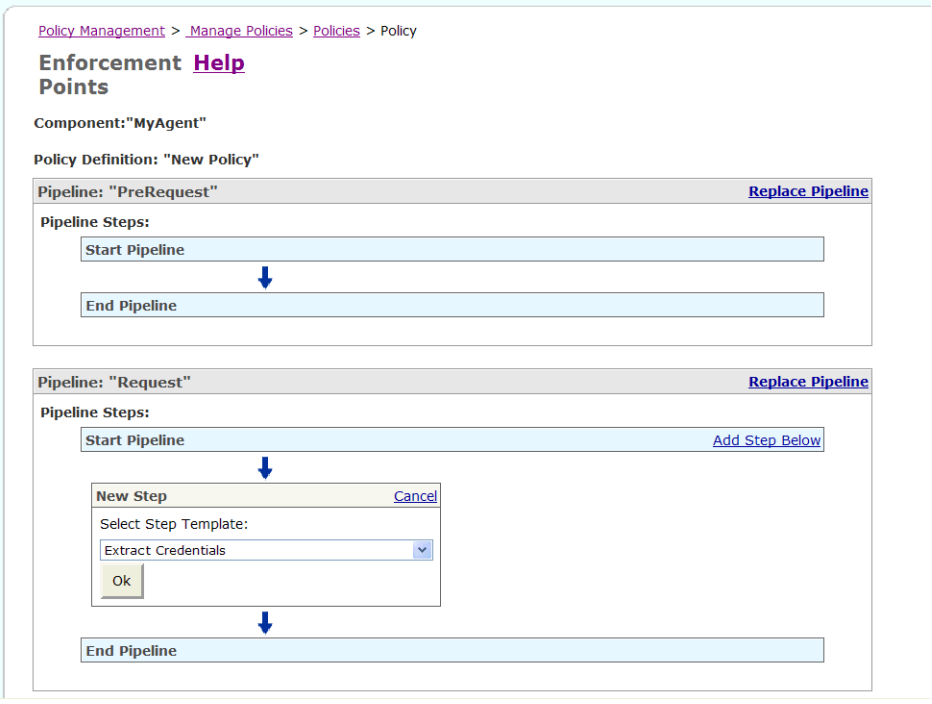
Figure 5–6 List of Policies for Oracle WSM Agent



3. You can either click the **Edit** icon to edit the Default Policy, or click **Add New Policy** to create a new policy.
4. In the pipeline where you want to add a policy step, click **Add Step Below.**

   The page refreshes and the New Step box appears (Figure 5–7).

Figure 5–7 Add Step Below Page with the Select Step Template List



5. Select a policy step from the Select Step Template list, and click **OK**.

The selected policy step is added to the pipeline.

6. Click **Configure** to configure the properties for this step.

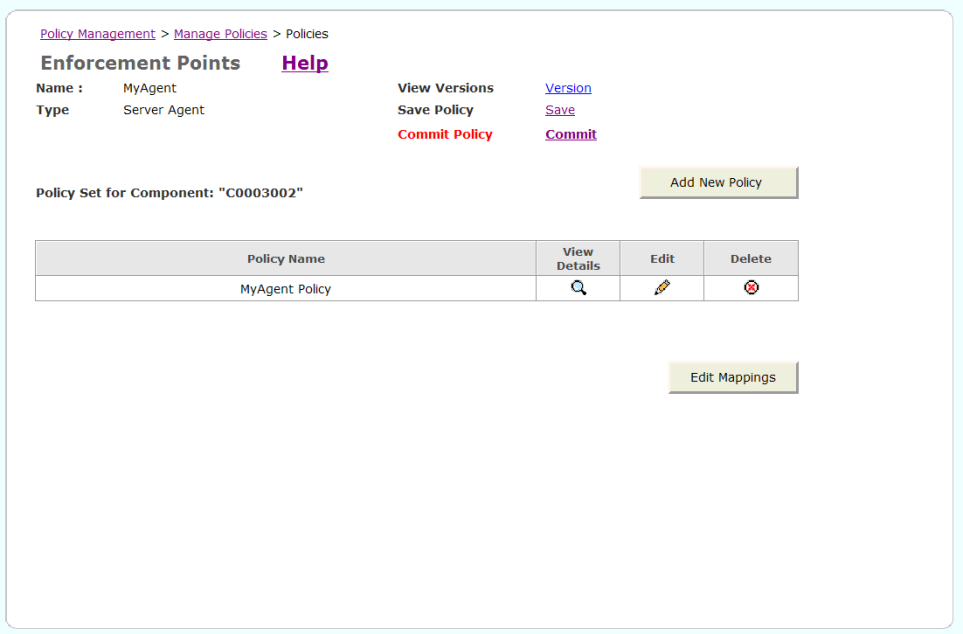
Note: Click the question mark (**?**) next to the name of the policy step for help on how to configure the properties.

Click **Environment Properties** to display a list of properties that appear in the form ${*propertyName*} and a description of the properties. You may use items from this list in setting properties in text fields. For example, for the XML Encrypt policy, you could specify the keystore environment property ${acme.keystore} in the **keystore location** field.

---

7. When you are finished editing the properties, click **OK.**
8. Continue to add and configure policy steps.
9. When you are finished, click **Next.**
10. Enter the name of the policy and click **Save**.
11. The Commit Policy field appears in red, alerting you to commit your changes (Figure 5–8). Click **Commit** to update the policy for the associated agent.

Figure 5–8 Committing Changes to the Policy



The *Policy is committed* message is displayed.

Note: You can click **Save** to save the policy to a file. You can then open this file to see the XML representation of the policy. Or you can configure the agent or gateway to run in disconnected mode and refer to this file for the policies, rather than run in connected mode where the **policy enforcement point (PEP)** connects to the Oracle WSM Policy Manager.

Assigning Policies to Web Service URLs for Agents

By default, the Default Policy is applied to all services hosted by Oracle Application Server. There may be situations where there are multiple services, and you want to assign different policies to each Web service. In this situation, assign the policy to the appropriate Web service.

To assign a policy to a Web service URL

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

   A list of registered gateways and agents is displayed.

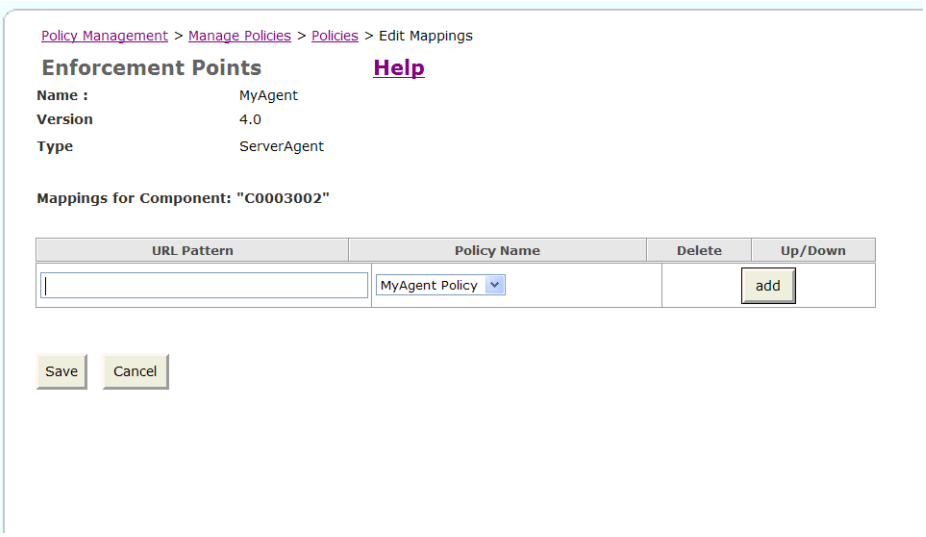
2. In the row for a specific agent, click **Policies**.

   Oracle WSM displays the policy set for the agent.

3. Click **Edit Mapping.**

   The Enforcement Points/Mappings page is displayed (Figure 5–9).

Figure 5–9 Assigning a Web Service URL to an Oracle WSM Agent



4. Enter the URL for the Web service to which this policy applies, and click **Save**.

   Oracle WSM displays a confirmation message to confirm that the mapping was successfully updated.

Configuring the Log Policy Step to Record SOAP Messages

The request and response pipelines of the default policy include a log step that causes policy enforcement points (PEP) to record SOAP messages to either a database or a component-specific local file. When you create a new policy to replace the default, you must configure the Log step in each pipeline to enable SOAP message recording.

You can configure logging for each policy by using one or both of the following:

- Log Step in the Request Pipeline – Logs information from the service request.
- Log Step in the Response Pipeline – Logs information from the service response.

The logging level can be configured in the Log step. You can choose from among the following log levels:

- Header – Only the SOAP header is recorded.
- Body – Only the message content (body) is recorded.
- Envelope – The entire SOAP envelope, which includes both the header and the body, is recorded. Any attachments are not recorded.
- All – The full message is recorded. This includes the SOAP header, the body, and all attachments, which might be URLs existing outside the SOAP message itself.

You can send the SOAP messages to the default Oracle WSM database or to a database dedicated just to SOAP messages.

In order to send SOAP messages to a file, you must specify the location of the log files using the cfluent.messagelog.file.logDirectory property of the policy enforcement point. Specify an absolute path location to the file. The log file is in Multipurpose Internet Mail Extensions (MIME) format. You can edit the cfluent.messagelog.file.maxFileSize property for the policy enforcement point and specify that the file be rotated automatically when it reaches a specified size. See "Making Changes to Your Policy Enforcement Points" on page 9-4 for more information on editing these properties.

Note: To transfer previously recorded SOAP messages from a local file to a database, use the data import tools provided by your RDBMS application.

Typically, system performance improves when log files are located in topological proximity to the enforcement component. Therefore, Oracle recommends multiple distributed logs in a highly distributed environment.

Log files are stored in a logs directory; each log file name includes the time that the log reached its maximum size and log messages that were written to the file. Set the location for the log files using the cfluent.messagelog.file.logDirectory property. See "Making Changes to Your Policy Enforcement Points" on page 9-4 for more information on editing this property.

Using Pipeline Templates

Pipeline templates are reusable policy pipelines. A pipeline template can be created for a specific section of a particular policy enforcement point. For example, you can create a pipeline template for the request pipeline for a server agent.

Pipeline templates allow you to apply consistent policy enforcement for Web services deployed across an enterprise.

Creating Policy Pipeline Templates

This section describes how to create a pipeline template.

To create a pipeline template

1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Pipeline Templates.**

   A list of the pipeline templates for the default choices, Gateway component and PreRequest pipeline, is displayed.

2. To add a new pipeline template, click **Add New Pipeline Template.**
3. Select the type of component for which you want to create a pipeline template from the Component Type list.
4. Select the pipeline for which you want to create a template from the Pipeline Type list.
5. Specify a name for the template in the **Pipeline Template Name** field.
6. Click **Next.**

   The Web Services Manager Control displays a second page where you specify policy steps for the new policy template.
7. Click **Add Step Below**.
8. Select a policy step from the Select Step Template list and click **Ok.**
9. Configure the properties for the policy.
10. Continue adding and configuring policy steps.
11. When you are through making changes to the pipeline template, click **Save,** then click **Ok.**

Using a Pipeline Template in a Policy

You may substitute sections of a pipeline template with previously created pipeline templates.

To use a pipeline template in a policy

1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Manage Policies.**
2. Click **Policies** for the gateway or agent whose policy you want to edit.
3. Click **Edit** for the policy you want to edit.
4. Click **Replace Pipeline**.
5. Choose the template you want to use from the New Pipeline Template list, and click **Replace.**

   The steps in the pipeline template are added to the policy.

   Note: If the pipeline contains any policy steps, these policy steps are deleted and replaced with the policy steps from the selected template.
6. Add, delete, and configure additional steps for the pipeline.
7. When you are finished editing your policy, click **Save.**
8. Enter a name for your policy, and click **Save**, then click **Commit**.

Restoring an Earlier Version of a Policy

Each time you make a change to a policy and commit the change, a new version of the policy is created. The most current version is the policy that is enforced. This policy is called the **working version.** If necessary, you can return to an earlier version of the policy and restore it to the working version.

To restore an earlier policy

1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Manage Policies.**
2. In the List of Components, click **Policies** for the component whose policy you want to restore.
3. In the View Version field, click **Version** to view the policy version history for the component.

   In the List of Policy Versions for Component: *component_ID*, the policy versions are listed in descending order. The *working version* is the same policy as the one that is identified as the *enforced version*.

   Click the **View** icon, then click the **View Details** icon to see the policy definition.
4. From the List of Policy Versions for Component: *component_ID* page, click **Restore** for the version you want to restore as the working policy.

   A message appears prompting you to confirm the change. Click **OK.**
5. Click **Ok.**

> **Note:** If you have added or deleted a Web service for the component, or changed the URL mappings to a later policy version, the policy in the previous configuration will no longer apply, and the policy cannot be automatically restored. To restore the policy, you will have to reassign the current URLs to the policy and enter it as a new policy version.

6. In the Commit Policy field, click **Commit** to commit the change.

   The message *Policy is committed* appears in the Commit Policy field.

Click **Version** in the View Version field to return to the policy version history. You will see that the version number of the enforced policy has increased by 1.

Purging Obsolete Versions of a Policy

A new version of the policy is created each time you commit a change to the policy. Over time, you can accumulate policy versions that have become obsolete. You can purge, that is, permanently delete, obsolete policies that you no longer need.

To purge obsolete policy versions

1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Manage Policies.**
2. In the List of Components, click **Policies** for the component whose policy versions you want to purge.
3. In the View Version field, click **Version** to view the policy version history for the component.
4. Click **Purge Policy Set Versions.**
5. In the *Purge all policy set versions prior to version* box, enter the policy version.

Note: All versions of the policy set up to the version you enter will be purged. For example, if you enter 25, then any versions that exist up to and including version 24 will be purged.

6. Click **Purge.**
7. A message appears asking you to confirm the purge. Click **Ok.**

   A message appears confirming that the purge was successful.

6

Monitoring Oracle Web Services Manager

From the Oracle Web Services Manager (Oracle WSM) Operational Management menu options, you can view and manage performance of your Web services.

This chapter includes the following sections:

Oracle WSM Monitor

The Oracle WSM Monitor collects metrics (statistical measurements, operational status, and other information about service execution) from several sources, for Web services managed by Oracle WSM. Oracle WSM Gateways send metrics to the Oracle WSM Monitor for each Web service managed through the gateway. Agents also send metrics to the Oracle WSM Monitor, whether they are embedded in the client or in the server. You can view the metrics collected by Oracle WSM Monitor through the Operational Management section of Web Services Manager Control.

The Oracle WSM Monitor does the following:

- Correlates and stores operational metrics on Web service execution and Web Services Manager component performance.
- Restricts data queries and views of system configuration information to authorized parties.
- Allows administrators to define rules for actions and conditions that generate alarms and notification.
- Allows administrators to create custom reports by recording performance of managed Web services and Web Services Manager components.

The following are the benefits of Oracle WSM Monitor:

- Improves application integrity by alerting information technology (IT) administrators of service-level performance or availability issues for any Web service in the network, so issues can be quickly addressed.
- Reduces time in troubleshooting and diagnosing problems by quickly identifying Web services and applications causing failures or performance problems.

- Enhances security by continuously monitoring the entire services network for security violations.
- Ensures conformance to service-level agreements (SLAs); administrators can be notified when service performance fall below set limits.

Operational Management Overview

The Operational Management options in Web Services Manager Control let you display snapshots of Oracle WSM system and Web service performance, create custom views of Oracle WSM metrics and performance data, and create alarm rules and view service alarms.

In the left pane of Web Services Manager Control, there are three distinct sections in the Operational Management menu—for performance reporting, creating and displaying customized views, and creating and viewing alarms. You can select the **Snapshot**, **My Views**, and **Alarms** options simply by clicking the corresponding links. Menu options that display the icon, that is, **Overall Statistics, Security Statistics, Service Statistics** and **Alarm Rules**, include submenu options, so you can click the main menu choice to expand and display the available submenu options.

Table 6–1 provides a brief description of each Operational Management menu option that appears in Web Services Manager Control.

Table 6–1 Operational Management Navigation Pane Options

Menu Option	Description
Snapshot	Provides an overall view of Oracle WSM managed services. Visual dashboard elements display overall, security, and service statistics for selected Oracle WSM components and services.
Overall Statistics:	Provides an expandable menu with choices that display overall Oracle WSM performance measurements.
SLA Compliance	Compares actual performance metrics against configurable SLA conformance limits.
Execution Details	Provides a graph of service execution attempts; also provides information on success or failure, latency, failover, and so on.
Message Logs	Lists message log entries for service requests and responses. For failures, you can drill down to view error messages.
Flows	Provides a searchable list of monitored flows. A **flow** is a collection of Web service invocations that are grouped together within some context, for example, a collection of application services required to fulfill a client request in processing an order.
Security Statistics:	Provides an expandable menu with choices that display Oracle WSM performance measurements pertaining to authentication and authorization.
Access Control	Shows distribution of access failures for all service execution attempts. Graph also shows the distribution of all granted requests and failures of authorization and authentication for all access attempts.
Service Statistics:	Provides an expandable menu with choices that display Oracle WSM performance measurements pertaining to service latency and message traffic.
Latency Variance	Provides a graph showing maximum and minimum latency metrics, by individual service or all services.

Table 6–1 (Cont.) Operational Management Navigation Pane Options

Menu Option	Description
Traffic Analysis	Displays number of bytes and messages per service by individual service or all services.
My Views	Provides access to customized views of service and message log information based on atomic (field value) or aggregated measurements; also allows you to create custom views.
Alarms	Search and display alarms based on the Component ID, time range, alarm type, or alarm severity.
Alarm Rules	Provides menu choices to create alarm rules and process Oracle WSM alarm information.
Alarm Creation	Creates or edits Oracle WSM rules to generate alarms based on specific conditions such as service execution or policy step processing, latency, and SLA conformance triggers.
Alarm Processing	Specifies alarm processing such as actions to perform when rules trigger an alarm.

The following sections describe operations performed by each Operational Management menu group.

Viewing System Performance

Web Services Manager Control provides a centralized location from which system administrators can accomplish the following performance monitoring tasks:

- View a summary snapshot view for system health of all services, or view the health of only selected services.
- View statistics for service-level agreement (SLA) compliance, execution details, message logs, and transaction flows.
- Edit and monitor conformance of performance SLAs.
- View security statistics for authentication and authorization failures.
- View service statistics for service latency variances.
- Analyze message traffic.

The Oracle WSM Monitor captures service detail information, status, and performance metrics from each service request or response handled by Oracle WSM gateways or agents. That information is then stored in the Oracle WSM Database, where it can be retrieved by requests to display status, statistics, snapshots, views, and so on. You can also create rules that will process information stored in the database to generate alarms and other messages that alert you to potential problems.

Oracle WSM System Snapshot

The operational snapshot view gives system administrators an overview of performance and availability for all managed Web services. If problems arise that affect performance of one or more Web services, these problems will be reflected in the Snapshot view. The graphs in that view display information for three categories:

Overall Statistics

- **Overall failures:** Shows the average rate of failure occurring in any of the steps of a Web service policy pipeline, or during the execution of a selected service.

- **Overall latency:** Show the average latency for execution for the selected services and corresponding policy pipelines

Security Statistics

- **Authentication failures:** Shows the average rate of failure in the authentication step of the policy pipelines for the selected services.
- **Authorization failures:** Shows the average rate of failure in the authorization step of policy pipelines for the selected services.

Service Statistics

- **Service failures:** Shows the average rate of failure occurring during execution of a selected service. Does not include failures occurring in policy steps.
- **Service latency:** Shows the average latency for execution of the selected services. If a failure occurs in any policy step prior to service execution, no data will be included for those services.

Viewing a System Snapshot

This section describes how to see a snapshot view of the Oracle WSM system.

To view a system snapshot

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** and then select **Snapshot**.
2. The Snapshot page is displayed.

Oracle WSM generates the summary values for all services by processing values from all Web services, or for all services connected through a single enforcement component (a gateway or an agent), as indicated by the Component list. The default view summarizes the data for all connected services.

Note: If you have just installed Oracle WSM, some dials similar to those in Figure 6–1 may not have any data to display. In that case, the dials will indicate an absence of data by a lighter color.

Viewing Service Performance

You can view all the services managed by one component, choose just one of the managed services, or designate a specific time range by choosing a combination of the **Component, Service,** and **Time Range** from the menus.

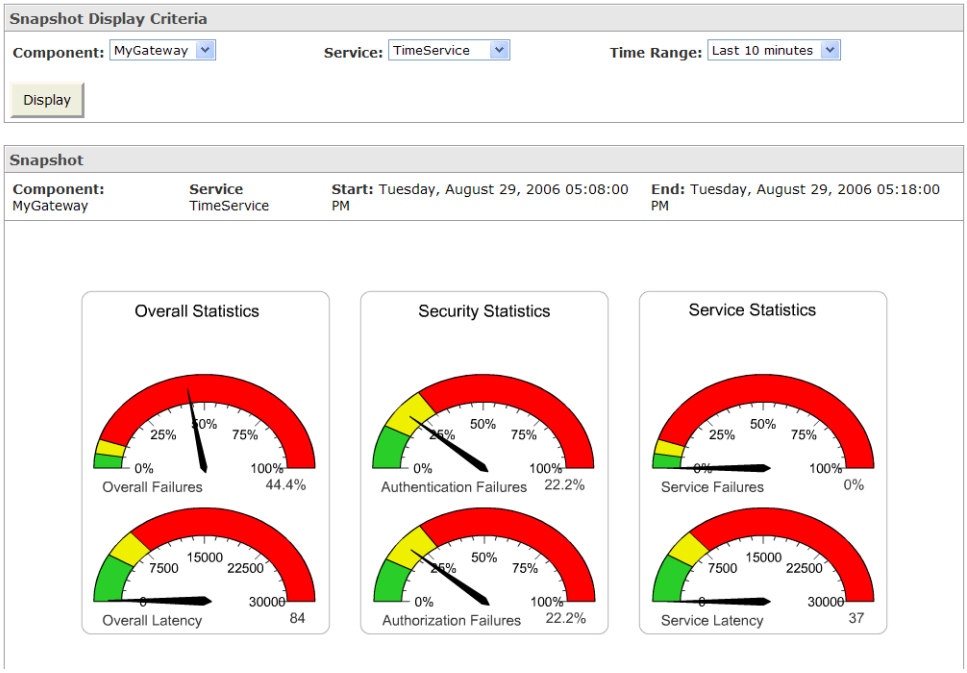
To view service performance

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** and then select **Snapshot.**
2. Select a component from the Component list.

   Web Services Manager Control now displays statistics and performance measurements specific to the component you selected.
3. From the Service list, select a specific Web service for which to display measurements. The list of services in the list are the services managed by the selected component.
4. From the Time Range list, select the time period for which to display results.
5. Click **Display**.

Web Services Manager Control now displays a snapshot based only on data for the selected component, service, and time range.

Figure 6–1 Oracle WSM System Snapshot



Oracle WSM Overall Statistics Options

In the navigation pane of Web Services Manager Control, click **Operational Management**, and then select **Overall Statistics** to generate real-time reports on Oracle WSM performance. These reports are based on statistics that combine measurements for both the policy execution and Web service execution. For example, the latency statistics combine the time taken to execute all the steps in the policy pipeline for a service (for example, logging, authentication, authorization, and decrypting,) with the time for the execution of the Web service.

The options for Overall Statistics reporting in Web Services Manager Control include the following:

- Viewing compliance with service-level agreements for all services or only for selected services
- Editing service-level agreements
- Viewing detail information on Web services execution
- Viewing message logs
- Viewing execution flows across multiple services

Service-Level Agreement Compliance

For all Web services managed by a gateway or agent, Oracle WSM creates default service-level agreement (SLA) settings that are in force until you edit them to define your own service performance requirements. You can edit an SLA configuration for any service registered to an Oracle WSM component.

Oracle WSM compares the values, derived from actual traffic metrics, with the SLA for each Web service. The distribution of values is displayed in red, yellow, and green zones on the bar graphs or pie graphs, according to the categories shown in Table 6–2:

Table 6–2 Graphical Display Color Key

Color	Meaning
Green	Web services performing within SLA
Yellow	Web services performing close to limit of SLA
Red	Web services performing outside limit of SLA

Monitoring SLA Compliance

This section describes how to use Oracle WSM to monitor compliance with service-level agreements for all services.

To monitor the service-level agreement (SLA) compliance for all services

- From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Overall Statistics,** and then select **SLA Compliance**.

  Web Services Manager Control displays a summary of SLA compliance for all Oracle WSM managed services.

Viewing SLA Compliance

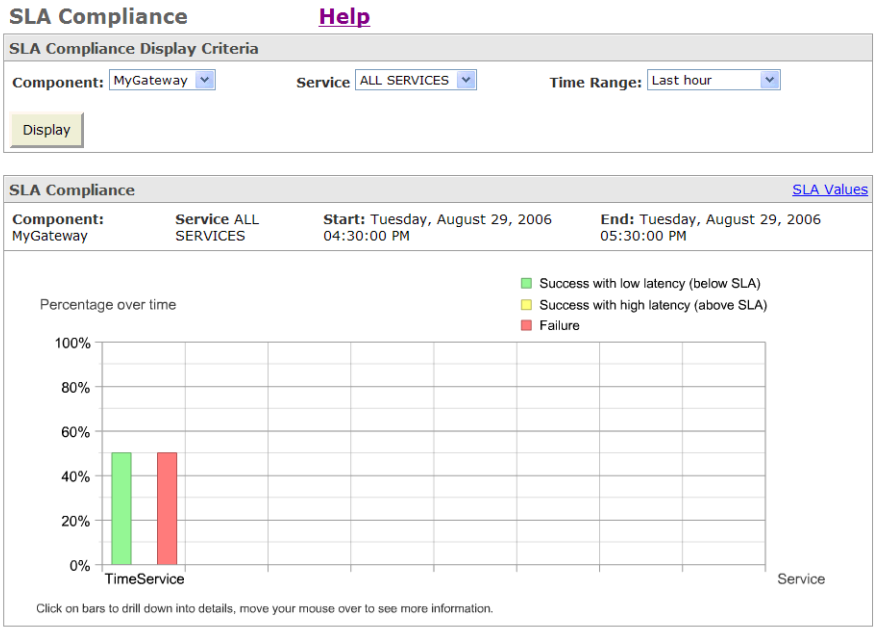
This section describes how to use Oracle WSM to monitor compliance with service-level agreements for a selected component, service, and time range.

To view SLA Compliance for a selected component, service, and time range

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Overall Statistics,** and then select **SLA Compliance**.

   By default, Oracle WSM displays the SLA Compliance for all services in the last hour.

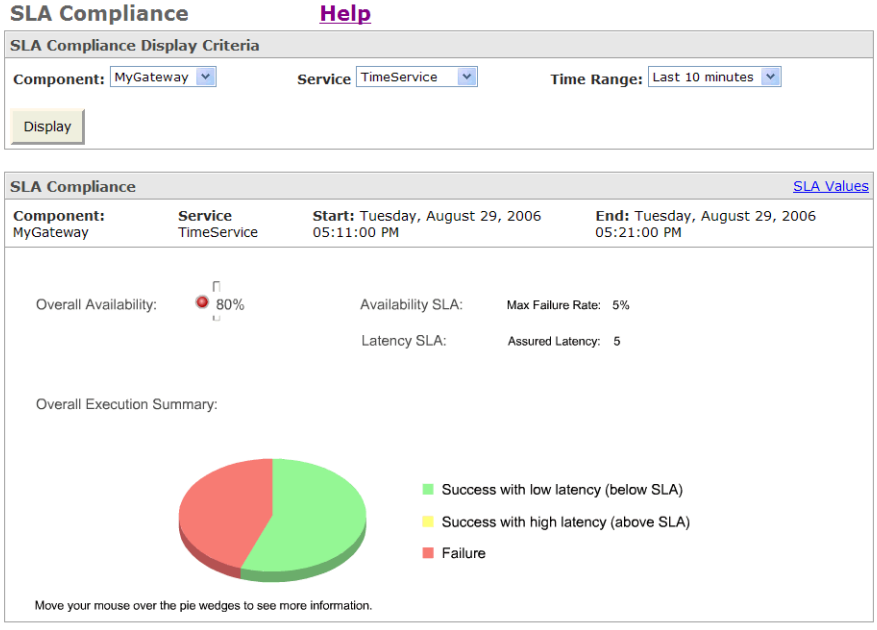
Figure 6–2 Service-Level Agreement (SLA) Compliance for All Services



2. From the display criteria lists, select the component, service, and time range for which you want to view SLA compliance.
3. Click **Display**.

   Web Services Manager Control now shows SLA compliance for the selected component, service, and time range.

Figure 6–3 Service-Level Agreement (SLA) Compliance for a Selected Service



From this display, you can move your cursor over particular sections of the pie chart to display more detailed information on the overall service execution.

Editing Service-Level Agreements

This section describes how to specify specific conformance levels for latency, downtime, and failure rates.

To edit specific conformance levels for latency, downtime, and failure rates

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Overall Statistics,** and then select **SLA Compliance.**
2. From the Component list, select a component.
3. Click **Display**.

   Web Services Manager Control displays overall statistics and SLA compliance for the selected component.
4. Click **Edit SLA Values**.

   Web Services Manager Control now lets you view and update specific SLA parameters.
5. Select the component and the Web service for which you want to edit SLA conformance levels.
6. Click **Next.**

   Web Services Manager Control displays current SLA limits defined for the specified component and Web service.
7. Specify the new SLA limits you want to enforce.
8. Click **Save.**

> **Note:** Oracle WSM uses default SLA values for any SLA limits for which you do not specify a value.

Service Execution Details

Execution Detail views that Oracle WSM provides give administrators a way to monitor the details of service invocations. Execution Detail views display instances (represented by dots) of actual messages over time, plotted against their overall latencies. The color of a dot indicates the overall status of a service execution.

Viewing Execution Details for All Services

This section describes how to view the execution details for all services.

To view execution details for all services

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Overall Statistics,** and then select **Execution Details.**

   By default, Web Services Manager Control displays execution details for all connected services.
2. To view more information on specific service execution instances, position your cursor over individual dots in the graph.

   When you position your cursor over a dot, Web Services Manager Control displays a brief message listing the service name, service ID number, time, and latency measurement for the execution instance.

Viewing Execution Details of Specific Services

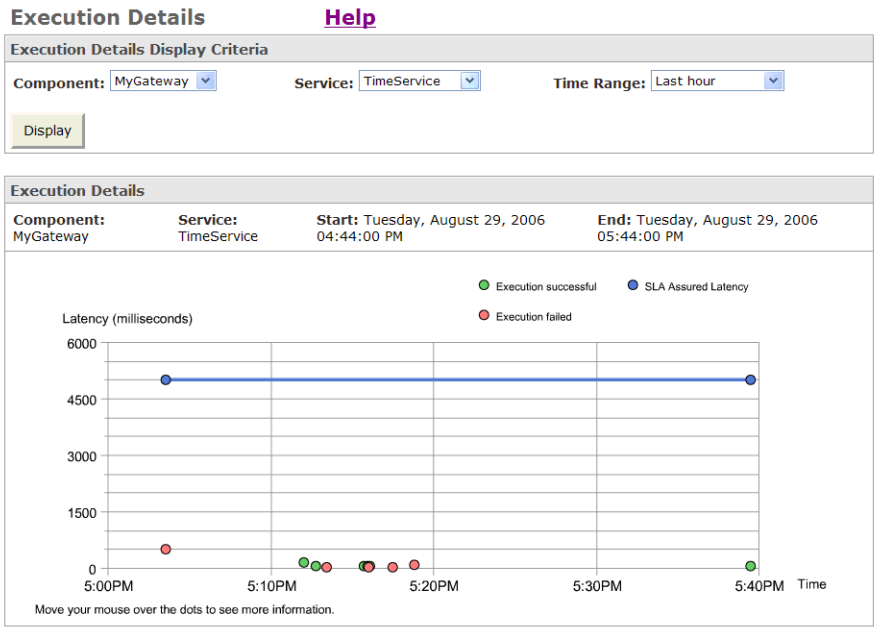
This section describes how to view the execution details for a specific Web service.

To view execution details of specific services

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Overall Statistics,** and then select **Execution Details.**
2. Select a component, service, and time range from the lists.
3. Click **Display.**

   Web Services Manager Control displays execution details only for the selected Web service.

Figure 6–4 Execution Details for a Specific Service



You can position your cursor over individual dots in the graph to view more information on specific service execution instances. Web Services Manager Control displays a brief message listing the service name, service ID number, time, and latency measurement for the specific execution instance.

Oracle WSM Message Logs

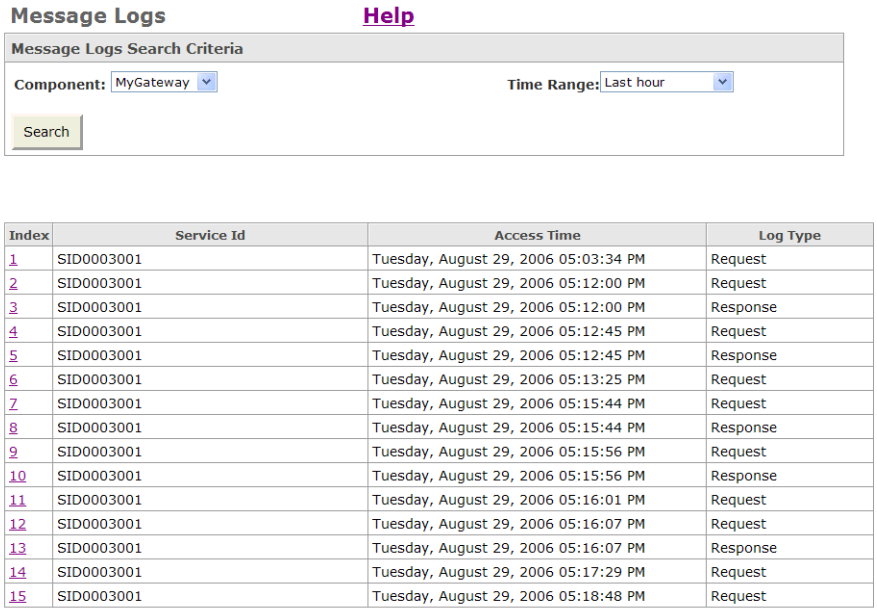
By default, Oracle WSM policies create logs for all request and response messages processed through gateways or agents. Message logging for a Web service is customized by modifying the properties for the logging step in the service's policy pipeline.

For more information on configuring the Log step properties for either gateways or agents, see "Adding Policy Steps to Oracle WSM Gateway Web Services" on page 5-9 and "Creating Policies for Oracle WSM Agents" on page 5-12.

To view message logs

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Overall Statistics,** and then select **Message Logs.**

Figure 6–5 Message Logs for Gateway



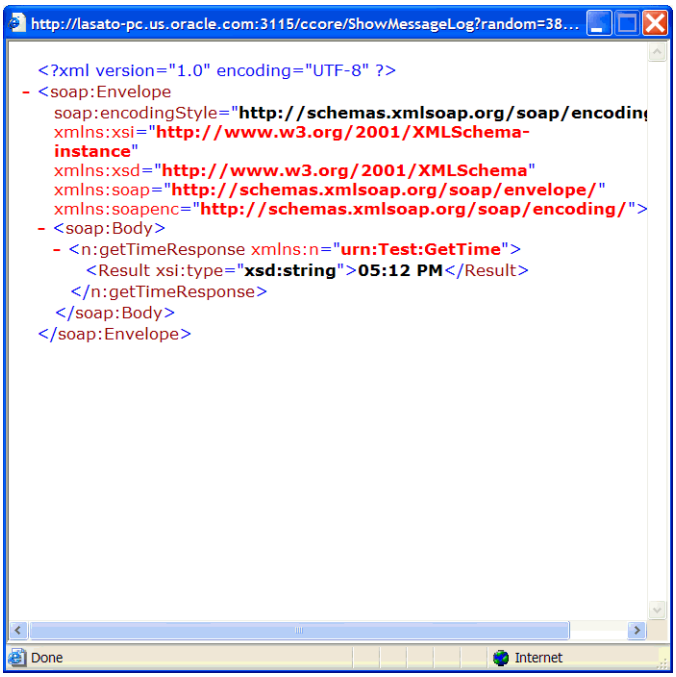
Index	Service Id	Access Time	Log Type
1	SID0003001	Tuesday, August 29, 2006 05:03:34 PM	Request
2	SID0003001	Tuesday, August 29, 2006 05:12:00 PM	Request
3	SID0003001	Tuesday, August 29, 2006 05:12:00 PM	Response
4	SID0003001	Tuesday, August 29, 2006 05:12:45 PM	Request
5	SID0003001	Tuesday, August 29, 2006 05:12:45 PM	Response
6	SID0003001	Tuesday, August 29, 2006 05:13:25 PM	Request
7	SID0003001	Tuesday, August 29, 2006 05:15:44 PM	Request
8	SID0003001	Tuesday, August 29, 2006 05:15:44 PM	Response
9	SID0003001	Tuesday, August 29, 2006 05:15:56 PM	Request
10	SID0003001	Tuesday, August 29, 2006 05:15:56 PM	Response
11	SID0003001	Tuesday, August 29, 2006 05:16:01 PM	Request
12	SID0003001	Tuesday, August 29, 2006 05:16:07 PM	Request
13	SID0003001	Tuesday, August 29, 2006 05:16:07 PM	Response
14	SID0003001	Tuesday, August 29, 2006 05:17:29 PM	Request
15	SID0003001	Tuesday, August 29, 2006 05:18:48 PM	Request

2. Select the component and time range from the lists.
3. Click **Search.**

   The Message Logs page displays a list of service requests and responses matching the specified component and time range. For each log entry, Web Services Manager Control displays the service ID, the time stamp of the log entry, and the type of log entry, request or response.

   From this display, you can click the hyperlinked number entries in the Index column to view the details of specific responses or requests corresponding to log entries.

Figure 6–6 Message Log



Note: The next section describes how you can view flow execution details from Web Services Manager Control. You can display the same log request or response detail as that shown in Figure 6–6, by selecting individual service invocation points on the graphs that show flow execution details.

Configuring the Number of Messages Displayed

When you display the message logs for a component, by default, Oracle WSM displays up to 500 of the most recent message logs. If there are more than 500 message logs, you will not be able to view messages 501 and beyond. You can configure the number of messages that are displayed by editing the following file:

ORACLE_HOME/owsm/config/ccore/ui-config-installer.properties

Edit the ui.messagelog.maxViewableMessageLogs property and specify the maximum number of messages you want displayed. By default, this property is set to 500.

ui.messagelog.maxViewableMessageLogs=500

If you do not want to set a limit, that is, you want to be able to see all messages, then specify zero (0), as in

ui.messagelog.maxViewableMessageLogs=0

Viewing Flow Execution Detail

A **flow** is a collection of Web services's invocation data that is grouped together within a particular context. For example, fulfilling an order may require access to several back-end Web services to complete the entire operation. All these service invocations for a single transaction are grouped together to define the context of an order fulfillment flow.

Note: For more information on designing message structures so that Oracle WSM can effectively interpret how messages are related in the same flow or transaction, see "Designing Messages Structures for Oracle WSM Flow Tracking" on page 6-12.

With options available from Web Services Manager Control, you can view the service execution flow across multiple Web services.To view flow execution from Web Services Manager Control:

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** then select **Overall Statistics,** and then select **Flows** option.
2. Select the flow type and time range from the lists.
3. Click **Search.**

   Web Services Manager Control displays a list of all service flows selected by your search criteria.
4. To see the details of a particular flow, click **Context ID**.

   Web Services Manager Control displays a summary of the different service invocations involved in the flow.
5. Click **Chart View**.

   Web Services Manager Control now displays a graph showing details of the service invocations in the flow. In the graph, the horizontal axis represents time (relative to flow execution) and the vertical dimension shows the services invoked in each of the flow instances.

   - The blue horizontal line represents the flow time line.
   - Each service invocation (or subflow) is denoted by a blue vertical line.
   - Each Web service (subflow type) has its own time line, represented by the horizontal lines.
   - Each section of a horizontal line is color-coded; **GREEN** represents successful execution, **RED** indicates a service failure, and **YELLOW** represents that the service was successful after failover.

Designing Messages Structures for Oracle WSM Flow Tracking

Flows are messages that are related to each other in some application-determined manner. Messages may be related because they are part of an overall transaction, or because they trigger each other. The Web service application determines which messages are related to each other, and indicates this by inserting the same correlation ID in all the messages that are part of the same flow. This correlation ID can be thought of as the flow ID.

How Does an Application Put the Flow ID into the Message?

The application developer who is writing the application that invokes a Web service must put flow IDs into the message headers. The same flow ID should be inserted into the headers of any related messages. Flow IDs are inserted into the SOAP header using the `<RelatesTo>` tag defined by the WS-Addressing specification, for example:

```
<soap:Envelope
        xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:wsa="http://schemas.xmlsoap.org/ws/2003/03/addressing"
```

```
      xmlns:cswm="http://schemas.confluentsw.com/ws/2003/08/Message">
  <soap:Header>
      <wsa:RelatesTo RelationshipType="cswm:ParentContext">
            uuid:8EB9-C6A3-75AA-7EBA
      </wsa:RelatesTo>
  <soap:Header>
  <soap:Body>  ......   </soap:Body>
 </soap:Envelop>
```

How Does the Oracle WSM Monitor Track Flows?

Oracle WSM agents and gateways automatically inspect each message that passes through them to see if the SOAP header contains a `<RelatesTo>` tag. If a `<RelatesTo>` tag is found, then the flow ID is extracted and sent along with other metrics about the message (latency, sender, destination, and so on) to the Oracle WSM Monitor.

The Oracle WSM Monitor stores the flow IDs for all the messages it receives and then correlates the messages containing the same IDs to deduce the flow.

How Does the Oracle WSM Monitor Know When a Flow Started and When It Ended?

Only the application being monitored can tell when the flow transaction is completed. The Oracle WSM Monitor provides SOAP APIs that allow applications to tell the Oracle WSM Monitor when the flow begins and when the flow ends. These SOAP APIs are, in effect, calls to the Oracle WSM Monitor that say:

```
monitor.beginFlow (flowId);
....
monitor.endFlow (flowId);
```

You do not *have to* send the begin and end SOAP messages to the Oracle WSM Monitor to view flows in the Operational Management section of Web Services Manager Control. Flows for which the Oracle WSM Monitor has not received begin and end messages show up as *pending* in the Flows view. If you choose to send the begin and end messages, you must use Java APIs that construct the SOAP message and send it to the Oracle WSM Monitor.

> **Note:** If you have specific requirements for tracking message flows in Oracle WSM and need more technical details on how to implement the changes, contact your Oracle support representative or Oracle Support Services.

Oracle WSM Security Statistics Options

The Security Statistics option provides a way for you to view and monitor authentication and authorization activity with Oracle WSM managed services. Authentication failure occurs when the proper credentials are not presented during enforcement of the Oracle WSM policy step for authentication, that is, when a user tries to invoke a Web service without first obtaining the proper permissions.

The Oracle WSM Security Statistics option provides two different graph type options to display access failures, as described in Table 6–3.

Table 6–3 Security Statistics Graph Types

Graph Type	Description
Bar Graph	Shows the distribution of access failures per Web service, per selected component.
Pie Graph	Represents per component, for all Web services registered to that component, and shows the distribution of all access attempts; total successes over failure of authorization and failure of authentication.

Viewing a Summary of Access Control Violations

By default, Web Services Manager Control displays a summary of security statistics access control failures.

To view a summary of access control violations

- From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Security Statistics,** and then select **Access Control**.

Viewing Selected Access Control Violations

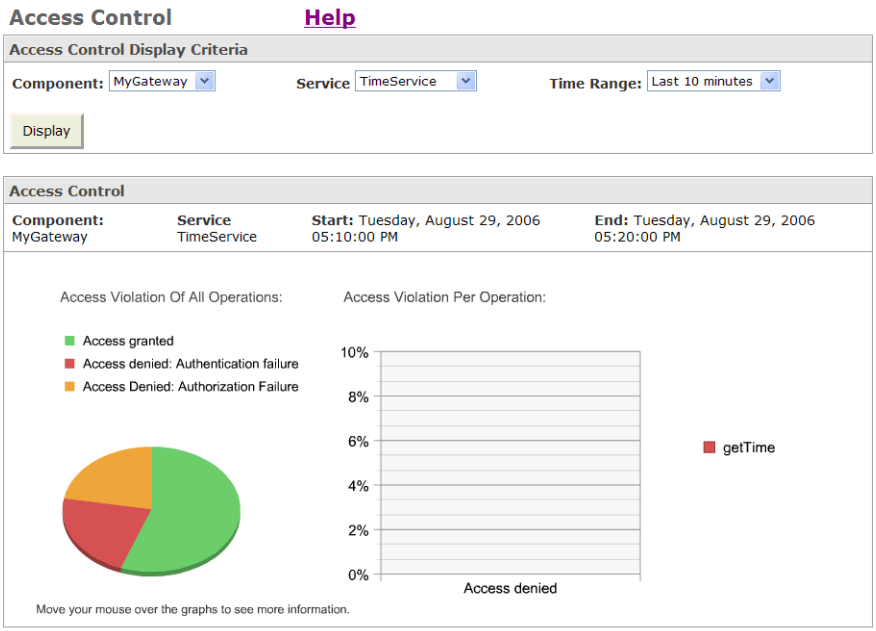
From Web Services Manager Control, you can also view access control violations for selected components, services, and time ranges.

To view selected access control violations

1. From the navigation pane of Web Services Manager Control, click **Operational Management,** select **Security Statistics,** and then select **Access Control**.
2. Select the specific component, service, and time range for which to view access control information.
3. Click **Display.**

   Web Services Manager Control now displays access control violations only for the component, service, and time range you specified.

Figure 6–7 Access Control for Selected Service



From this display, you can move your cursor over particular sections of the pie chart to display more detailed information on access control attempts and failures. Web Services Manager Control displays a window showing the actual percentages of access attempts, successes, and failures.

Oracle WSM Service Statistics Options

The Service Statistics options in Web Services Manager Control (Latency Variance and Traffic Analysis) show statistics based solely on service execution, not including the latency of any policy steps. If the service fails in the policy execution, prior to the service execution, no statistics will be displayed.

Viewing Service Latency Variance

Web Services Manager Control provides an option to view service latency variance, which allows administrators to monitor the variations in Web service performance over a selected time period.

To view service statistics for latency variance

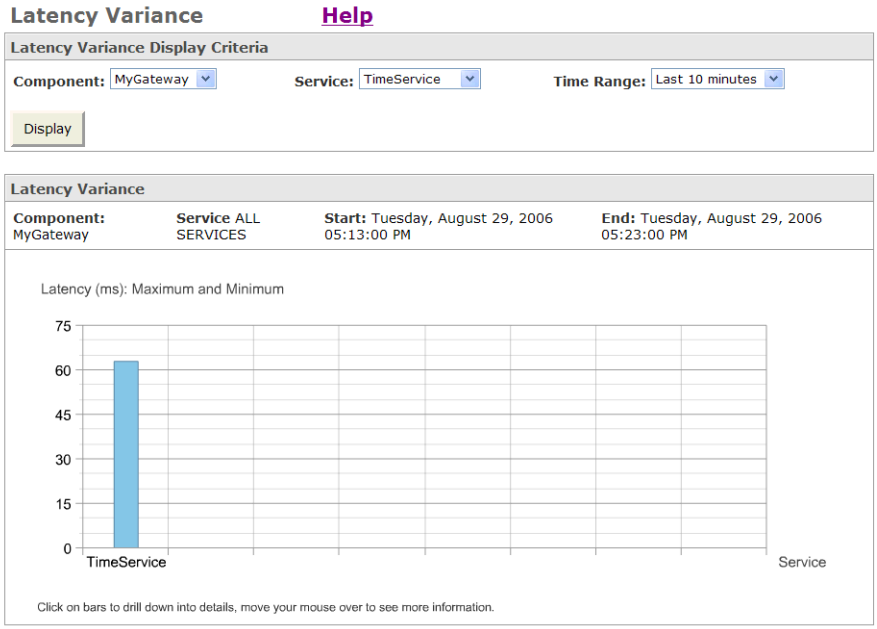
1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Service Statistics,** and then select **Latency Variance**.
2. The latency variance for all services is displayed.

Viewing Latency Variance for Selected Display Criteria

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Service Statistics,** and then select **Latency Variance**.
2. Select the desired component, service, and time range for which you want to view latency variance information.
3. Click **Display**.

Web Services Manager Control now displays the latency variance for the selected component, service, and time range.

Figure 6–8 Latency Variance for Selected Service



Traffic Analysis

The Traffic Analysis option provides an analysis of the volume of message activity, including message size, for selected services.

To view traffic analysis statistics

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Service Statistics,** and then select **Latency Variance**.

   Web Services Manager Control displays charts reflecting message traffic for all selected services.

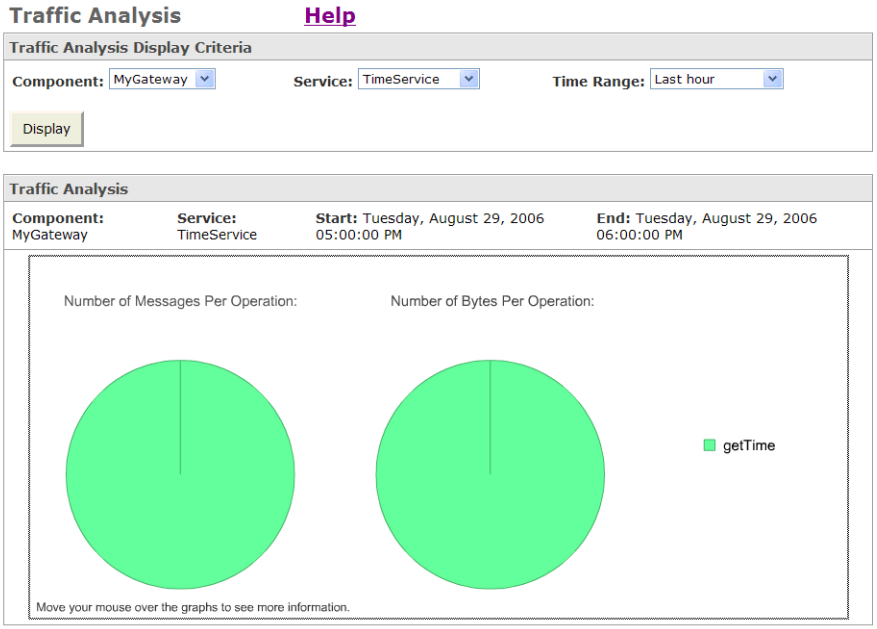
   From this display, you can move your cursor over particular sections of the pie chart to display more detailed information on individual services. A brief message will display information on the number of messages per operations.

To view traffic analysis for selected display criteria

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Service Statistics,** and then select **Traffic Analysis**.
2. Select a desired component, service, and time range from the lists.
3. Click **Display**.

   Web Services Manager Control displays message traffic information for the selected component, service, and time range.

Figure 6–9 Traffic Analysis for Selected Service



As before, you can move your cursor over particular sections of the pie chart to display more detailed information on individual services. In this case, a brief message will display information on the number of messages per operation.

Creating Custom Views of Oracle WSM Metric Displays

The Snapshot, Overall Statistics, Security Statistics, and Service Statistics options in the Operational Management menu provide standardized views of metrics that Oracle WSM monitors and captures for system and service operations. Using the Operational Management My Views option, you can create customized views of metrics and other Oracle WSM data. In creating a customized view, you define and save a template that defines a set of data and, optionally, a time interval, to capture and display metric information. You can then go back anytime later to run one of your views and display the metrics and other information that matches its parameters and conditions.

Oracle WSM lets you define any number of customized views. For example, you can create separate views that focus the data on metrics and other service detail information captured by the Oracle WSM Monitor for particular gateways, agents, and services. You can also specify different views that either aggregate data field values (the average over several metrics) or return all *atomic* (single-event metric) data sent to the Oracle WSM Monitor from selected gateways, agents, and services.

Selecting Customized Views to Display Metrics

Note: Views are created and maintained on a per group basis.

To display and select from a list of customized views

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** and then select **My Views**.

Web Services Manager Control displays the My Views page that shows a list of any existing customized views that you have created and saved.

From this page, you can choose to run an existing view, delete or edit a view, or create a new customized view.

2. Click the **Run** icon.

   When you run a view, Web Services Manager Control displays the results for the selected view.

   In the aggregated view, you see performance statistics such as the number of total requests, successes and failures, the average overall and service execution latency, plus some other related metrics. You can click the hyperlinked headings in the table to sort the rows, based on data values in a particular column. In addition, you can click the hyperlinked data values in columns showing either average or total (count) values to drill down and view the detailed service metrics that were included in the calculation.

   In a nonaggregated view, you see the metrics for specific service requests (and responses), with specific atomic data values in each row. In addition, you can click the hyperlinked index column values to display the details of the request or response log entry for specific operations.

   Oracle WSM displays up to 25 rows of service request data per page. You can click **Previous, Next,** or a specific hyperlinked **Page Number** to navigate pages contained in the view.

Creating a New Custom View

You can create as many customized views of Oracle WSM metrics as you want. Views are maintained and displayed on a per group basis.

To create a new view

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** and then select **My Views**.
2. From the My Views page, click **Create New View**.

   Web Services Manager Control displays a New View form in which you can specify a name for the view and select options that will determine the metrics to be included in the view.

   Oracle WSM shows a different list of display fields depending on whether you selected **Atomic** (single-event metric) or **Aggregated** (counting or averaged over several metrics). Table 6–4 and Table 6–5 provide descriptions of the different columns available in the two views.

   Fields that require an entry on the New View form are denoted with an asterisk (*). To begin, specify a name for the view and then also specify the components and associated services for which you want metrics to be included in the view. In both the Components and Services lists, you can select one or more components or services (using Shift or Control keys to select multiple items in the list).

   In the Time fields, you can optionally specify a start time, and end time, or both a start and end time, to define an interval of time by which to qualify metrics included in the view. If you specify only a start or end time, you can optionally also specify a duration that will designate the metrics to be included in the view, for a period of time following the start time, or before the end time. In specifying start and end times, you can choose keyword options such as -2 hours or -2 days

before the current time (relative to the time the query is run). Or, you can enter discrete start and end dates and times using the format: MM-dd-yyyy HH:mm. The time zone field lets you specify the time zone in which the start and end times will be calculated, so the view will include metrics generated only in the specified time period.

Table 6–4 Display Field Options for Atomic (Single Event) Views

Field	Description
Component	Gateway or agent name where message was processed.
Service	Name as service being invoked (last part of URL).
Operation	Service operation that was invoked.
Overall Status	Overall status of the invocation, including policy and service execution.
Authentication Status	Success or failure indicator if authentication step in policy is attempted.
Authorization Status	Success or failure indicator if authorization step execution in policy is attempted.
Service Status	Success or failure, failover, or pending indicator if service execution is attempted.
Size	Total size of request and response messages, if information is available. (Size is currently available only for messages handled by gateways.)
Overall Latency	Overall time delay of a service invocation, including policy and service execution.
Service Latency	Overall time delay of service execution.
User ID	User ID of the user invoking the service, if available.
Timestamp	Time of invocation (when request message arrived at gateway or agent).

Display fields listed in Table 6–4 are available when you choose the Atomic Display Fields option to create an Atomic (nonaggregated) view. Display fields listed in Table 6–5 are available when you choose the Aggregated Display Fields option to create an aggregated view.

Table 6–5 Display Field Options for Aggregate Views

Field	Description
Component	Gateway or agent ID where messages were processed.
Service	Service being invoked (displays last portion of service request or response URL).
Operation	Service operation that was invoked.
Number of Requests	Total number of request and response messages processed.
Number of Successes	Total number of messages successfully executed.
Number of Failures	Total number of messages where execution failed.
Number of Authentication Failures	Total number of authentication failures (for messages where authentication step is included in policy).
Number of Authorization Failures	Total number of authorization failures (for messages where authorization step is included in policy).

Table 6–5 (Cont.) Display Field Options for Aggregate Views

Field	Description
Number of Service Failures	Total number of service execution failures.
Average Overall Latency	Average latency of all service invocations (including policy and service execution).
Average Service Latency	Average latency of all service executions.
Number of Bytes	Total (aggregate) size of all request and response messages. (Size is currently available only for messages handled by gateways.)

For both types of views, you can select check boxes next to fields you want to appear and clear those check boxes you do not want to include in a view.

Viewing Oracle WSM Alarms and Defining Alarm Rules

The 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. By creating rules in Web Services Manager Control, you can select and process certain information to generate alarms and other messages that alert you to existing or potential problems with services managed by Oracle WSM. The Operational Management menu in Web Services Manager Control includes options to do all of the following:

- Search and view generated alarms based on the Component ID, time range, alarm type or severity.
- Create and edit rules to generate alarms based on specific conditions, such as service execution or policy step processing, latency, and SLA conformance triggers.
- Specify alarm processing and notification actions to perform when rules trigger an alarm.

The following sections describe options available to view generated alarms and how you can define new alarms and alarm processing rules for the operation of your Oracle WSM system.

Oracle WSM Alarms

As an administrator, you will want to be notified as soon as possible if there is a problem with one of your Web services. Oracle WSM has a system for creating alarms to alert you when a Web service is not functioning as planned.

This section includes the following topics:

- Viewing generated alarm data
- Creating new alarm rules
- Creating alarm processing rules

Oracle WSM ships with one default alarm for latency. To generate alarms, you must enable the default alarm or create a new alarm. Once enabled, generated alarms are logged and may be searched by component, time range, alarm type, and severity level.

Viewing Generated Alarm Data

The alarms menu lists all generated alarms. Oracle WSM ships with one default alarm rule. The rule is disabled until you change the setting. To generate visible alarm data,

you must enable the default alarm or create a new alarm. Each alarm type has a rule that is triggered by a specific condition. The conditions that trigger the rules can be reviewed by looking at the details of the alarm type for each rule. For more information, see "Creating an Alarm Rule with Oracle WSM" on page 6-21.

To view generated alarm data

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** and then select **Alarms.**

   The alarm browser displays a list of alarms, up to the first 25 most recent alarms. View the results beyond the first 25 by clicking **Get Next** until the results for the desired time period are displayed.

   From this page, you can also search and display specific alarm types by selecting **Component, Time Range, Alarm Type**, and **Severity** options from the lists and clicking **Search**.

2. View individual alarm details by clicking the associated service ID number.

   Web Services Manager Control displays details of the selected alarm message.

Creating an Alarm Rule with Oracle WSM

An alarm rule specifies how Oracle WSM processes metrics to trigger alarms. Once created, alarm rules are combined with a rule action, typically an alert sent to key personnel.

To create an alarm

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Alarms,** select **Alarm Rules**, and then select **Alarm Creation.**

   Web Services Manager Control displays alarm creation rules you have defined.

2. Click **Add New Rule.**

   Web Services Manager Control displays a page in which you can define a new alarm rule.

3. Enter the Alarm Type and enter a brief description.

4. Select a **Measurement Type** from the list.

   Alarm rules can be based on the measurement types described.

Table 6–6 Invocation Measurement Types

Measurement Type	Description
agginvocation	Statistics that summarize a set of invocation statistics received for a given service over a specified period of time (default =2 minutes). For example, using this type, you could create an alarm if average latency drops below a certain level.
aggping	Aggregated ping statistics. For example, using this type, you could create an alarm if a service is not accessible for 5 minutes.
correlated-flow	Flow events that happen within the same context. For example, using this type, you could create an alarm when a flow execution fails or takes more than a day to be completed.
flow-event	The start and end of a flow execution. For example, using this type, you could create an alarm when a transaction starts or ends.

Table 6–6 (Cont.) Invocation Measurement Types

Measurement Type	Description
invocation	Statistics for individual Web service executions. For example, using this type, you could create an alarm if service latency exceeds a specific threshold.
ping	Statistics based on synthetic requests. For example, using this type, you could create an alarm every time a service is not accessible.

The alarm rule action is automatically enabled.

5. Click **Next.**

   Web Services Manager Control now updates the page to let you specify an alarm severity level and conditions for the alarm rule to be triggered.

6. Choose an **Alarm Severity Level** from the list.

7. Specify a time zone and, optionally, specify days of the week and time intervals to qualify specific days and times when the alarm rule can be triggered.

   For the Time Intervals field, you can specify a time interval using the time format shown (hh:mm:ss).

8. Specify one or more individual rule conditions in the *Add new expression* field to define when the new alarm rule will be triggered.

   For example, you could define a condition, **latency** > **300**, by selecting the latency option from the list in the left column, select the right angle bracket (**>**) as the condition operator from the middle column, and specify **300** as the value in the right column.

   Note: A rule condition consists of one or more logical expressions connected by the Boolean operator AND. You can define logical expressions based on any of the fields or variables defined in the measurements type. Commas can be used to specify multiple valid values that can satisfy a rule condition, for example, status=0,1.

   For more information on creating rules, "Operational Rule Basics" on page 6-22.

9. Click **Save.**

   Web Services Manager Control updates the alarm creation rules list, now including the alarm rule you just created.

   The new Alarm rule is enabled and will be triggered by the conditions that match the specified rule condition. Generated alarms appear when you select **Operational Management,** and then select **Alarms** in Web Services Manager Control.

Operational Rule Basics

A rule consists of a condition and an action. The condition is a Boolean expression, and the action is an operation to execute when the rule is triggered. A rule can be seen as a conditional processing operation--if the condition is evaluated as true, then the action is performed.

Oracle WSM ships with a default set of metrics processing rules, and actions based on those rules. Rules may be assigned to a specific Web service, or they can operate over

all Web services for a component. Rules can be defined and immediately assigned to a Web service, or they can be saved and activated at a later time. Rules become operational only after they are enabled and committed.

There are two basic types of Oracle WSM rules, outlined Table 6–7:

Table 6–7 Types of Operational Rules

Rule Type	Description	Advantages	Limitations
General	Examines all metrics and takes action based on a single event.	Preserves information about specific events.	Could cause avalanche of actions in cases of Web service failure and have negative impact on system performance.
Aggregated	Examines only aggregated metrics and takes action based on a consolidated view of events.	Lowers the risk that alarms can overwhelm system performance.	Could lose information about specific events could during the aggregation process.

If it is important that a system administrator be notified each and every time a Web service fails, you can define a general operational rule. Even if the Web service fails very rarely, this rule runs for every incoming Web service invocation metric. Defining operational rules as general rules could have a negative impact on system performance.

If efficient run-time performance is a priority, you can define a rule based on aggregated metrics, and base your rule on the number of failed invocations for a specified period of time. Oracle WSM then gathers data during the time interval and evaluates the aggregated data immediately after the specified time interval has passed.

Available Statistics for Building Rules

Aggregated rules, as described in Table 6–7, perform calculations involving *integer statistics* elements, which provide a way of keeping statistics about the values of an atomic variable over the specific aggregation provided by the rule.

Integer statistics are recorded for the following elements:

- min: smallest value recorded during the aggregation.
- max: largest value recorded during the aggregation.
- last: last value recorded during the aggregation.
- count: number of values taken into account during the aggregation.
- total: sum of all values.
- average: total or count of all values or events.

  For instance, if the variable looked at is the speed of a car during a road trip and the aggregation has taken place over the whole trip, you might assemble the following statistics to describe the trip:

  - min : lowest speed recorded
  - max : max speed recorded
  - last : current speed
  - count : number of samples
  - total : total sum of all speed sample (not very relevant)

– average : average speed over the trip

When defining rules, Oracle WSM references elements from aggregated statistics using the following conventions:

```
variable_name@stat.min
variable_name@stat.max
variable_name@stat.last
variable_name@stat.count
variable_name@stat.total
variable_name@stat.average
```

Similarly, Oracle WSM references aggregated statistics that contain *buckets* which are conditional counters, using the following convention:

```
variable_name@bucket.bucketname
```

The following tables provides descriptions of statistics used in different types of rule aggregations.

Table 6–8 Agginvocation Rule Conditions

Name	Type	Description
latency-stats	integer statistics	Statistics about recorded overall latency, including both the policy execution and service invocation.
service-latency-stats	integer statistics	Statistics about recorded endpoint service latency.
message-stats	integer statistics	Statistics about the number of bytes exchanged in invocations.
service-latency-buckets	integer buckets	Invocation information in three buckets (or categories): ■ Number of successful invocations whose latencies are above expected value ■ Number of successful invocations whose latencies are below expected value ■ Number of failed invocations
access-control-buckets	integer buckets	Access control information in three buckets (or categories): ■ Number of granted invocations ■ Number of denied invocations because of a failed authentication ■ Number of denied invocations because of a lack of permissions
component-id	string statistics	Statistics about the ID of the component that issued this measurement. (Used for display and storage indexing purposes.)
service-id	string statistics	Statistics about the ID of the invoked service. (Used for UI and storage indexing purposes.)
service-name	string statistics	Statistics about the ID of the invoked service. (Used for display and storage indexing purposes.)
operation	string statistics	Statistics about the name of operation (method) invoked on the service. (Used for display and storage indexing purposes.)

Table 6–9 Aggregated Ping Rule Conditions

Name	Type	Description
latency-stats	integer statistics	Statistics about recorded latency.
service-latency-stats	integer statistics	Statistics about recorded endpoint service latency.
service-id	string statistics	Statistics about the service ID of the pinged service. (Used for display and storage indexing purposes.)
service-name	string statistics	Statistics about the service ID of the pinged service. (Used for display and storage indexing purposes.)
url	string statistics	Statistics about the pinged URL. (Used for display and storage indexing purposes.)
component-id	string statistics	Statistics about the service ID of the component that issued this measurement. (Used for display and storage indexing purposes.)
latency-buckets	integer buckets	Invocation information in three buckets (or categories). 1: Number of successful pings whose latencies are above expected value. 2: Number of successful pings whose latencies are below expected value. 3: Number of failed pings.

Table 6–10 Flow-Event Rule Conditions

Name	Type	Description
type	string	The type of business process.
event-id	integer	Event flows; can be of two types: 1: Flow level events 2: Task level events
status	integer	The status of the current event: 0: Success 1: Failed 2: Pending

Table 6–11 Ping Rule Conditions

Name	Type	Description
service-id	string	The ID of the invoked service.
component-id	string	The ID of the component generating the measurement.
pinged-url	string	The URL used to retrieve the WSDL document.
status	integer	A successful invocation indicating that the WSDL document was returned. 0: Invocation successful 1: Invocation failed

Table 6–11 (Cont.) Ping Rule Conditions

Name	Type	Description
status-change	integer	The dynamics of multiple ping operations. This element indicates if the service went up, went down or did not change its status from the previous ping operations. Should be used to define alarm rules. 0: no-change 1: went-up 2: went-down
latency	integer	Latency, in milliseconds, of WSDL document retrieval.
http-status	integer	Status of the HTTP connection. (The current implementation uses the HTTP protocol to retrieve WSDL documents.)

Table 6–12 Invocation Rule Conditions

Name	Type	Description
flow-id	string	The ID of the service flow.
error message	string	Explanation describing why a service invocation failed.
service-id	integer	Unique ID assigned to service by Oracle WSM upon service registration.
authentication-status	integer	Status of the user authentication: 0: Authentication successful 1: Authentication failed
service-latency	integer	Endpoint service latency, in milliseconds.
service-status	integer	Endpoint service status: 0: Invocation successful 1: Invocation failed 2: Invocation failed over; the primary service end point failed, but a failover end point was successful. 3: Invocation pending. The service was contacted, data was sent to it; but no reply has yet come back.
size	integer	Number of bytes exchanged for both query and response.
user-id	string	The ID of the user that accessed the service.
component-id	string	The ID of the component that issued this measurement.
operation	string	Name of the operation (method) invoked for the service.
latency	integer	Overall latency, in milliseconds, including both the policy execution and service invocation.
authorization-status	integer	Status of the authorization: 0: Authentication successful 1: Authentication failed

Table 6–12 (Cont.) Invocation Rule Conditions

Name	Type	Description
status	integer	0: Invocation successful
		1: Invocation failed
		2: Invocation failed over. The primary service end point failed, but a failover end point was successful.
		3: Invocation pending. The service was contacted, data was sent to it, but no reply has yet come back.

Creating a Processing Rule for an Alarm

Alarm processing rules enable you as the administrator to receive an automatic alarm notification, or to trigger remedial actions. Once defined, processing rules for alarms are used to trigger actions (typically notifications of various types) that are sent to key personnel. The action desired is indicated by the alarm processing rule.

Oracle WSM ships with four default alarm processing rules that are disabled until you complete the information needed for the alarm action. You may also create your own alarm processing rule.

To create the action that the alarm takes when it is triggered

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** select **Alarms,** and then select **Alarm Processing**.

   Web Services Manager Control displays the default list of alarm processing rules that ships with Oracle WSM.

   You can add alarm processing rules to this list. For example, you could create a processing rule to trigger another alarm or action.

2. Click **Add New Rule.**

   Web Services Manager Control displays a new page in which you can define and specify actions for a new alarm processing rule.

3. Enter a name by which you can identify the new alarm processing rule and a brief description of the rule.

4. Select **Alarm Type** from the list.

5. Specify a **Rule Action** from the list, that is: e-mail notification, SNMP V1 Event, SNMP V2 Event, Service Invocation, or Custom.

6. Click **Next.**

   Web Services Manager Control now updates the page to let you specify conditions for the processing rule action to be performed.

7. Select the **Enabled** check box to enable the alarm processing rule.

8. Specify a time zone and, optionally, specify days of week and time intervals to qualify specific days and times when alarm processing rules can be triggered.

   For the Time Intervals field, you can specify a time interval using the time format shown (hh:mm:ss).

9. Specify a condition to trigger the action of the alarm processing rule.

   You can generally enter a condition in the same way you specify conditions for an alarm creation rule. You could use this condition to process a generated alarm differently for different services or components. For example, different

administrators could be notified when latency alarms are triggered for different services, depending on the service owner.

> **Note:** A rule condition consists of one or more logical expressions connected by the Boolean operator AND. You can define logical expressions based on any of the fields or variables defined in the measurements type. Commas can be used to specify multiple valid values that can satisfy a rule condition, for example, status=0,1.

10. Click **Next.**

    Web Services Manager Control displays a page for entering e-mail address and mail message information.

11. Enter the e-mail address for notification, and select or copy a desired message to include in the e-mail message body. The message body can be composed and can include alarm measurement variables, for example, latency value. If you pick the rule action SNMP V1, you will have to define how to compose ID and value fields from the available measurement variables.

12. Click **Save.**

    The list of alarm processing rules is displayed with your new alarm processing rule action added to the list.

> **Note:** If you send an e-mail notification when an alarm is triggered, you must configure Oracle WSM with the E-mail Server and user account information. See Chapter 3, "Configuring the Oracle Web Services Manager Components," in *Oracle Web Services Manager Deployment Guide* for more information.

Configuring Metrics Data Persistence

You can view Oracle WSM metrics over a period of time. For example, you can get overall statistics for a gateway over the last 2 hours or for the last 30 days. You can select a time period from the Time Range list. By default, the Oracle WSM Database persists data only for the last 100 minutes. So, if you select the time range **Last 10 minutes** or **Last hour** from the Time Range list, you will see the metrics for the period you selected. If, however, you want to see metrics over a longer period of time, you must configure the Oracle WSM Database to store the data for the desired length of time.

To configure the time period for which metrics data is persisted

1. Edit the following file:

   ORACLE_HOME/owsm/config/coreman/monitor-config-installer.properties

2. Change the value for the `monitor.aggregator.measurementStore.WindowSize` parameter.

   The number specified is the length of time, in minutes, that the Oracle WSM Database retains metric data.

3. Redeploy the application by using the `wsmadmin deploy monitor` command.

7

Managing Oracle Web Services Manager Roles

This chapter describes how to configure Oracle Web Services Manager (Oracle WSM) roles.

This chapter includes the following sections:

> **Note:** The concepts of groups and roles do not apply to Oracle WSM when it is installed as part of Oracle Application Server 10*g* Release 3 (10.1.3.1.0).

Managing Oracle WSM Access and Permissions

To control access to Oracle WSM components and operations, you assign user groups defined in your environment to Oracle WSM administrative roles. First, assign the Oracle WSM administrative roles to the groups defined and stored in your database or maintained in your LDAP server. Then, specify which groups are authorized to administer individual components (gateways and agents) and individual Web services managed by Oracle WSM.

> **Note:** For details on configuring Oracle WSM to use either the Oracle WSM Database or your own LDAP server to manage users and groups, see "Configuring the Oracle WSM Authentication Source" on page 7-5.

Table 7–1 Oracle WSM Roles

Role	Description
Super User	This is the primary Oracle WSM role, whose group members are responsible for the Oracle WSM site installation and deployment. Super User group members can access all features and perform all operations for any administrative component, PEP, or managed service. This includes adding, editing, or deleting components and their associated services. Only one group can be assigned the role of Super User. The Super User can delegate administrator responsibility for an installation to the Domain Administrator role and can assign user groups to other Oracle WSM roles.
Domain Administrator	The group assigned to this role is typically responsible for the day-to-day operations and management of an Oracle WSM system. Domain administrators have the same access rights as the Super User, and they can perform the same operations on all components and managed services. Domain Administrators can assign users to any role, except Super User and Domain Administrator. Only one group can be assigned the role of Domain Administrator. A Domain Administrator typically delegates administrator responsibility for individual components to groups assigned the Component Administrator role.
Component Administrator	Groups assigned to the Component Administrator role are given administrator responsibility for a specific component. Component Administrators directly administer or delegate administrator responsibility for managed services associated with the component. Component Administrators can edit the details of a component as well as add, edit, or delete managed services and routing associated with a component. They cannot, however, add, delete or remove the component nor can they change the registration details for a component.
Component Support	Groups assigned to the Component Support role are given support access to a specific component. Users assigned to the Component Support role can view information about the assigned component and its associated services. They are not allowed to add, edit, or change details of either the components or its associated services.
Service Administrator	Groups assigned to the Service Administrator role are given administrator responsibility for a specific service. Service Administrators can view and edit details and policy steps of the service. They cannot change service registration details, delete the service, or add new services.
Service Support	Groups assigned to the Service Support role can access a specific service. Users assigned to this role can view details and policy steps of the service. They are not allowed to edit service details, add or delete services.

When a user logs in to Web Services Manager Control, Oracle WSM automatically maps the group to which the user belongs with the role to which the group is assigned. Oracle WSM then determines what component and managed Web services that user is allowed to access and administer. For example, a user belongs to the group CSV.Admin.Gateway1, which is mapped to an Oracle WSM Administrator role and is given access to a gateway component. This user can automatically view and administer that gateway and its associated services when he or she logs in to Web

Services Manager Control. At the same time, other users belonging to groups assigned to roles with fewer permissions will be restricted in the operations they are able to perform on the same gateway and its associated services.

Assigning the Super User Role

Before you can assign groups within your organization to Oracle WSM roles, you must first assign a group to the role Super User. This group must be added to the Oracle WSM Database by logging in to the Oracle WSM Database and executing the following SQL command:

```
INSERT INTO GROUP_ROLE_MAPPINGS VALUES ('group_name',1,'Y')
```

The variable *group_name* is the name of the group to which you want to assign the role Super User.

Once this group has been added, members of this group can assign a group to the Domain Administrator role and other Oracle WSM roles. See"Assigning Oracle WSM Roles" for more information.

Assigning Oracle WSM Roles

One of the first tasks the Super User should perform after installing Oracle WSM is assigning a group to the Oracle WSM Domain Administrator role. The Super User or Domain Administrator can then assign other groups to roles that administer or support individual components or managed services. It is important that users assigned to Oracle WSM Super User and Domain Administrator roles be familiar with the group roles and user membership existing within their own organization, *before* they start to assign those groups to Oracle WSM roles.

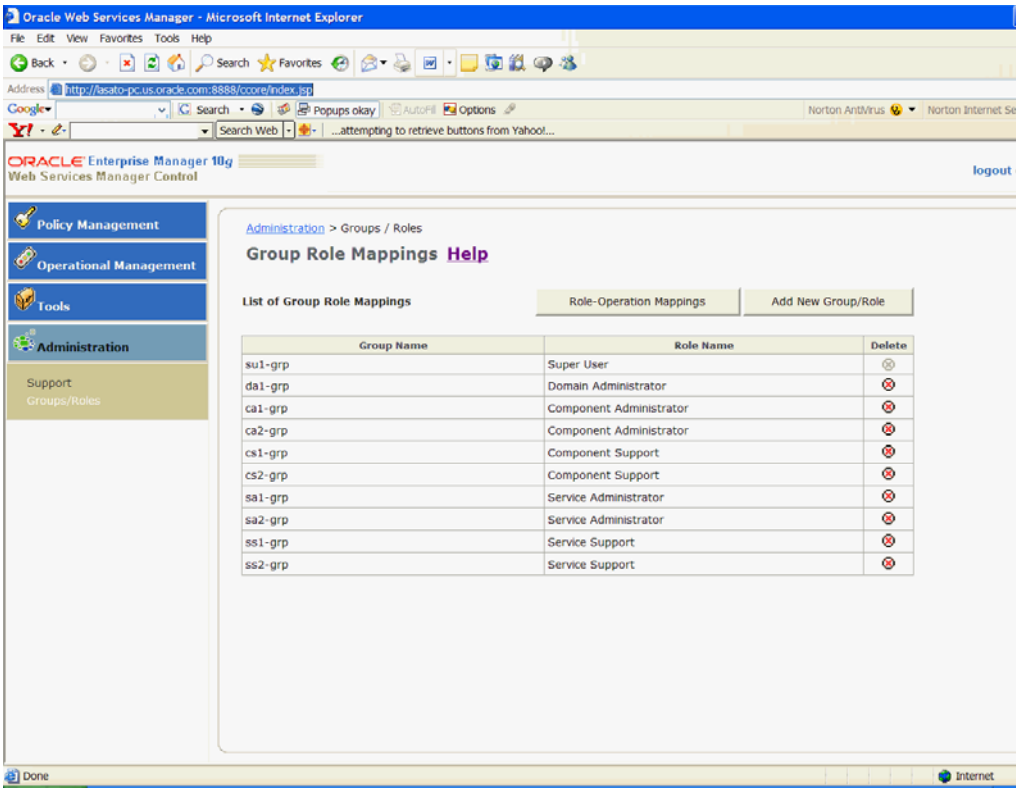
To assign Oracle WSM roles

Only users logged in as Super User or Domain Administrator can assign groups to roles. Component Administrators and Component Support users can view the role assignments, but they cannot remove or add new assignments.

1. Log in to the Oracle WSM Web Services Manager Control as Super User or Domain Administrator.
2. Click **Administration,** then click **Groups/Roles.**

   In the example (Figure 7–1), the List of Group Role Mappings page shows a group already assigned to the Domain Administrator role. The page also shows different groups assigned to component administrator, service administrator, and service support roles. If you assign multiple groups to the same role, this allows you to distribute the management of components and services amongst multiple groups.

Figure 7–1 Group/Roles Mappings Page

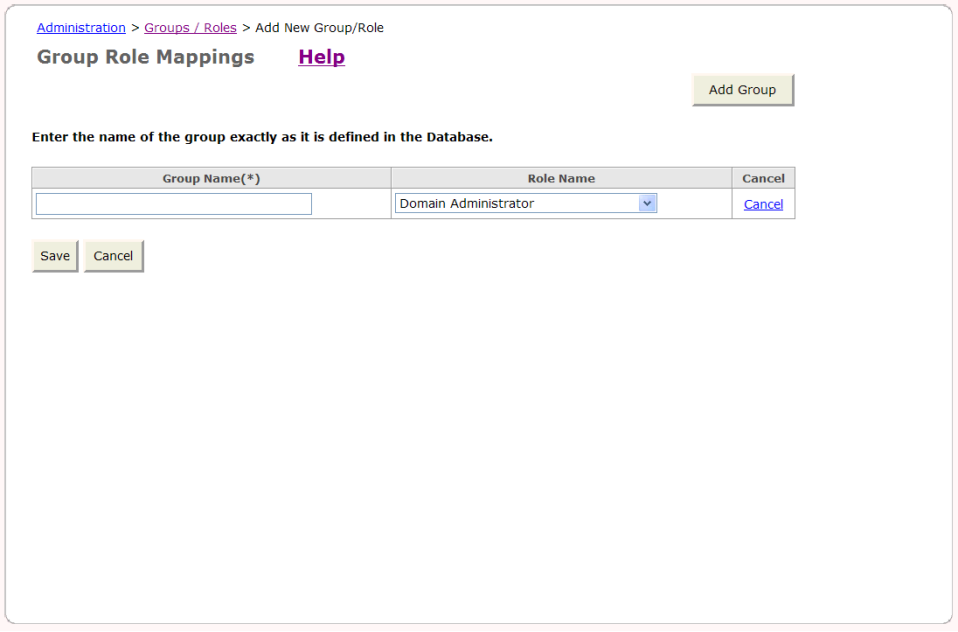


3. To add a new group/role mapping (assignment), click **Add New Group/Role**.

   At the top of the page (Figure 7–2), there is the following instruction: Enter the name of the group exactly as it is defined in the *source*.

   The term *source* is replaced with either the word *Database* or *LDAP Repository*. This instruction indicates whether your installation is using groups stored in a database or in an LDAP server.

Figure 7–2 Add New Group/Role Page



4. Enter the name of the group in the Group Name field, and select a role from the **Role Name** list.
5. Assign groups within your organization to Service Administrator and Service Support roles.
6. Click **Save.**

After groups have been assigned to roles, Domain Administrators will be able to add Component Administrator and Component Support group access to individual components when adding or editing the registration details of a component by selecting **Policy Management,** and then selecting **Manage Policies.**

In the Component Groups section, the group of the currently logged-in Domain Administrator will automatically appear in the list of groups having access to the new component. (The list is on the left side in Component Groups section.)

When logged in as a Domain Administrator, you can add additional groups allowed to access the new component with modify or view-only permissions. To do that, select one ore more groups from one or more of the lists on the right side of the display, then click **Add.**

Similarly, Component Administrators (as well as the Super User and Domain Administrators) can add Service Administrator and Support group access to individual services when adding or editing details of a service. To do that, select **Policy Management,** and then select **Register Services.** When you create a new service or edit the details of an existing service, use the Service Groups section to specify group access.

Configuring the Oracle WSM Authentication Source

You can configure Oracle WSM to use either a database or an LDAP server to manage users and groups by editing the *ORACLE_HOME*/owsm/config/ccore/ui-config-installer.properties file.

The user and group management parameters are defined in the section labelled *UI authentication properties.* This section provides default settings for two different authentication methods:

- Database
- LDAP server

The properties file includes the following parameters for the database:

```
ui.authentication.provider=com.cfluent.accessprovider.sampledb\
.LocalDBAuthProvider
ui.authentication.provider.properties=\
dbConnectionUrl=jdbc:oracle:thin:@sunserver5:1521:CCORE|\
dbDriver=oracle.jdbc.driver.OracleDriver|\
dbUser=cfluentdev|\
dbPassword=cfluentdev|\
maxConnections=10;\
idleTime=300;\
maxConnectionTime=120;
```

Table 7–2 Database Authentication Source Properties

Property	Description
dbConnectionUrl	Valid Java Database Connectivity (JDBC) connection URL.
dbDriver	JDBC driver class used to connect to the database.
dbUser	User ID of the schema owner for Oracle WSM Database.
dbPassword	Password for the user specified by dbUser.
maxConnections	Maximum database connections that are created. Default is 10.
idleTime	Ignore this parameter. It is obsolete.
maxConnectionTime	Ignore this parameter. It is obsolete.

The properties file includes the following parameters for the LDAP server:

```
ui.authentication.provider=com.cfluent.accessprovider.ldap\
   .BasicLdapAuthProvider
ui.authentication.provider.properties=\
ldapHost=dbserv1;\
ldapPort=389;\
ldapDN=ou=People,dc=corp,dc=confluentsw,dc=com;\
superUserRole=SystemAdmin;\
roleAttribute=groupmembership
```

Edit the parameters for the method you want to use, and comment out the parameters for the method you do not want to use.

Table 7–3 LDAP Server Authentication Source Properties

Property	Description
ldapHost	Host name of the system where the LDAP server is running.
ldapPort	Port on which the LDAP server listens for requests

Table 7–3 (Cont.) LDAP Server Authentication Source Properties

Property	Description
ldapDN	LDAP distinguished name (DN).
superUserRole	Group that is assigned the Super User role.
roleAttribute	Attribute for the user object that stores the groups (roles) to which the user belongs.
superUser	LDAP group that is assigned the Super User role in Oracle WSM.
roleAttribute	LDAP attribute name that identifies the user in the LDAP group.

After you have made your changes to the ui-config-installer.properties file, you must use the `wsmadmin deploy control` command for the changes to take effect. For more information on deploying applications, see *Oracle Web Services Manager Deployment Guide*. Once you have installed Oracle WSM with the user group configuration settings you want to use, Oracle WSM uses the specified source whenever it requires user authentication.

Note: Although you specify the LDAP group that is assigned the Super User role in Oracle WSM, you must manually add this group to the Oracle WSM Database.

Default Users and Groups

When you install Oracle WSM, the Oracle WSM Database is initialized with predefined users and groups that are assigned Oracle WSM roles. You can use these predefined groups and roles to test and stage an Oracle WSM installation, prior to deployment in a production environment.

Table 7–4 is a list of the default users, groups, and the roles to which they are assigned that are populated in the default installation.

Table 7–4 Default Users, Groups, and Oracle WSM *Roles*

Oracle WSM Role	Group	Users
Super User	`su1-grp`	`admin, su1.a`
Domain Administrator	`da1-grp`	`da1.a, da1.b`
Component Administrator	`ca1-grp`	`ca1.a, ca1.b, ca1.cs2.a, ca1.sa2.a`
Component Administrator	`ca2-grp`	`ca2.a,ca2.b`
Component Support	`cs1-grp`	`cs1.a, cs1.b`
Component Support	`cs2-grp`	`cs2.a, cs2.b, ca1.cs2.a`
Service Administrator	`sa1-grp`	`sa1.a, sa1.b, sa1.ss2.a`
Service Administrator	`sa2-grp`	`sa2.a, sa2.b, ca1.sa2.a`

Table 7–4 (Cont.) Default Users, Groups, and Oracle WSM *Roles*

Oracle WSM Role	Group	Users
Service Support	ss1-grp	ss1.a, ss1.b
Service Support	ss2-grp	ss2a, ss2.b, sa1.ss2.a

By default, the password for all predefined users is *oracle*. User names with designations such as `ca1.cs2.a` indicate that a user is a member of more than one group, and each group is assigned a different role. For example, the user `ca1.cs2.a` is a member of `ca1-grp`, which is assigned a Component Administrator role. The same user is a member of `cs2-grp`, which is assigned a Component Support role. The user's permissions is the combination of the roles assigned to groups to which the user belongs.

Manage Users and Groups Command

If you are using an Oracle WSM Database for user authentication, Oracle WSM provides a command-line tool to create users and groups. After creating your users and groups, and assigning users to groups, you can then assign roles to your new groups using the WSMADMIN command-line tool.

The command-line tool can be found in the following location:

ORACLE_HOME`/owsm/bin`

The syntax of the `wsmadmin` command to manage Oracle WSM users and groups is the following:

```
wsmadmin manageUserGroups [option]
```

The available options are shown in Table 7–5:

Table 7–5 manageUserGRoups Command Options

Option	Description
addUser	Adds a single user.
addGroup	Adds a single group.
addUserGroup	Adds an existing user to a group.
deleteUser	Deletes an existing user.
deleteGroup	Deletes an existing group.
deleteUserGroup	Deletes a user from a group.

This command-line tool can be used for adding and deleting users and groups only from the sample database shipped with Oracle WSM. If you store users and groups in an LDAP server, you cannot use this tool to add or remove users and groups. You must use whatever tools are provided with your LDAP server to perform these operations.

Configuring the manageUserGroup Properties File

When the command is executed, Oracle WSM checks the manageUserGroups.properties file that contains database connection information and information about the particular users and groups you want to add or delete.

The *ORACLE_HOME*/owsm/bin/manageUserGroups.properties file contains the following properties (Table 7–6) that you should set, based on the specific user or group operation you want to perform.

Table 7–6 manageUserGroups Properties File Settings

Property	Value
db_url	Specify the database URL to connect to the Oracle WSM Database. By default, this property is set to: jdbc:polite4@localhost:3120:orawsm
db_driver	Specify the driver used to connect to the Oracle WSM Database. By default, this property is set to: oracle.lite.poljdbc.POLJDBCDriver
db_user	Specify the database administrator user name to log in to the Oracle WSM Database. By default, this property is set to: system
db_password	Specify the corresponding password for the database user login name. By default, this property is set to: manager
user_id	Specify a user ID to uniquely identify a user. By default, this property is set to: ctang The user_id property must be specified for the addUser, addUserGroup, deleteUser, and deleteUserGroup operations.
user_name	Specify a login user name for the associated user_id. By default, this property is set to: Administrator The user_name property is required only for addUser operations.
user_password	Specify a password for the associated login user name and user_id. By default, this property is set to: oracle The user_password property is required only for addUser operations.
user_email	Specify the e-mail address to be associated with a specific user. By default, this property is set to: admin@admin.com The user_email property is required only for addUser operations.
group_id	Specify a group ID to uniquely identify a group. By default, this property is set to: IT-INFR The group_id property is required only for addGroup, deleteGroup, and deleteUserGroup operations.

Table 7–6 (Cont.) manageUserGroups Properties File Settings

Property	Value
group_desc	Specify a descriptive label for the associated group_id . By default, this property is set to: IT Infrastructure The group_desc property is required only for addGroup operations.

The db_password and user_password properties are specified when Oracle Web Services Manager is installed. The values are obfuscated and the manageUserGroups.properties file is populated with these obfuscated values. The user_password property is entered in the manageUserGroups.properties file as unencrypted text. Once the Oracle WSM Database is updated with the user information, this sensitive information should be deleted from the file.

Executing the wsmadmin manageUserGroups Command

Each time you want to perform an operation to add or remove users or groups, or add or delete users in groups, specify a new set of entries in the manageUserGroups.properties file. Then run the wsmadmin manageUserGroups command with the appropriate command-line option. For more information on the wsmadmin command, see *Oracle Web Services Manager Deployment Guide*.

8

Logging Events with Oracle Web Services Manager

This chapter describes the information collected by Oracle Web Services Manager (Oracle WSM). This chapter includes the following sections:

Overview

The Oracle WSM components generate the following types of information:

- Low-level **event** and state logs
- High-level performance metrics

Low-Level Event and State Logs

Each Oracle WSM component sends event execution and component status information to a local file dedicated to just that component. Such information is useful for monitoring the performance of your Oracle WSM system or debugging components that are not acting as expected. At various levels, this feature reports when the Oracle WSM Policy Manager starts or when a specific method is being executed, and it records a stack trace when certain errors occur.

The application log files can be found in the following location:

ORACLE_HOME/j2ee/*instance*/log

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*.

There, you will find the log files shown in Table 8–1:

Table 8–1 Oracle WSM Application Log Files

Log File Name	Contains Log Messages for
ccore.log	Oracle Enterprise Manager 10*g* Web Services Manager Control
coreman.log	Oracle Web Services Manager Monitor

Table 8–1 (Cont.) Oracle WSM Application Log Files

Log File Name	Contains Log Messages for
gateway.log	Oracle Web Services Manager Gateway
policymanager.log	Oracle Web Services Manager Policy Manager
clientagent.log	Oracle Web Services Manager Client Agents
serveragent.log	Oracle Web Services Manager Server Agents

By default, when you install Oracle WSM, the log files for the four components, Web Services Manager Control, Oracle WSM Monitor, Oracle WSM Gateway, and Oracle WSM Policy Manager are automatically created. The log files for the Oracle WSM agents are created only if you create and deploy client or server agents.

The maximum size of the log file is 10 megabytes, by default. When the log file reaches this maximum, a *1* is appended to the end of the file and messages continue to be logged to the application log file. For example, ccore.log would be renamed ccore.log.1, and messages would continue to be logged to ccore.log. By default, one backup file is created. Therefore, when ccore.log is renamed ccore.log.1, it overwrites any existing backup file with the name ccore.log.1.

Configuring Logging

You can configure the following characteristics of the log files: location and name of the log file (File), log level (priority value), log file size (MaxFileSize), and number of backup log files (MaxBackupIndex), by specifying the parameter in the parentheses in the logging.xml configuration file. Oracle WSM uses the Log4J technology developed by the Apache Group. Refer to the product documentation (`http://logging.apache.org`) for more information on configuring the log file parameters.

After you change any of the parameters in the logging.xml file, you must redeploy the application for the changes to take effect. See *Oracle Web Services Manager Deployment Guide* for information on deploying applications.

Logging Configuration Files

Each application has its own logging configuration file, named logging.xml, which you can use to configure logging. The location of the configuration files is as follows:

- The logging configuration file for all Oracle WSM components (*except* Oracle WSM Agents) is found at:

  ORACLE_HOME/owsm/config/*application*/logging.xml

  The variable *application* is the name of the Oracle WSM application: `ccore, coreman, gateway, policymanager.`

- The logging configuration file for OC4J agents (that is, server agents for an Oracle Web service and J2EE client agents) can be found at:

  ORACLE_HOME/owsm/config/interceptors/*component_ID*/config/*agent_type*/logging.xml

  The variable *component_ID*, is the component ID that is assigned to the agent when it is registered, and *agent_type* is `clientagent` or `serveragent`, specifying whether it is a client or server agent.

Changing the Log Level

You can control the amount of low-level event and state information sent to file by specifying one of the following hierarchical logging levels:

- **SEVERE –** Only events serious enough to cause a component failure are reported. Typically, these events are accompanied by a stack trace.
- **WARNING –** Program activity that has not caused program termination, but which could potentially affect system performance or data integrity.
- **INFO –** Confirmation of events that are supposed to occur (such as component startup, database connection, and so on). By default, the log level is set to INFO.
- **ALL –** All events logged with SEVERE, WARNING, and INFO log levels are reported.

> **Note:** These values are case-sensitive and must be specified in uppercase.

The log files allow you to specify a different logging level for each class. The following entries from the logging.xml file for Web Services Manager Control show the logging level set to INFO:

```
<category name="com.cfluent.webui">
<priority value="INFO" class="com.cfluent.ccore.util.logging.LogLevel"/>
</category>
<category name="com.cfluent.pingengine">
<priority value="INFO" class="com.cfluent.ccore.util.logging.LogLevel"/>
</category>
<root>
<priority value="INFO" class="com.cfluent.ccore.util.logging.LogLevel"/>
<appender-ref ref="FILE"/>
</root>
```

The logging level is specified with the `priority value` parameter. In most instances, you will not need to configure different log levels for specific classes. To change the log level for the entire component, do a search and replace to change the priority level for all classes to the desired level.

To change the log level for all Oracle WSM components except the third-party agents, see "Logging Configuration Files" on page 8-2. To change the log level for third--party agents, see the "Changing the Log Level for Third-Party Agents" section.

Changing the Log Level for Third-Party Agents

Third-party agents are deployed in containers other than OC4J (for example, IBM WebSphere or BEA WebLogic). The procedure for changing the logging level for these agents is different from other Oracle WSM components. For third-party agents, the runtime configuration file for these agents is modified. Generally, Oracle recommends that you do not modify runtime files because these settings are overridden if and when the component is redeployed. However, in the case of third-party agents, it is cumbersome to edit the source configuration files and reinstall the agents for what is generally a temporary change to the log level while a problem is being debugged. Therefore, in this limited instance, Oracle recommends changing the log level in the runtime configuration files as described in the following:

- **Server Agents** – The logging configuration file for server agents deployed on third-party containers is found in a directory where the Web service application is

deployed. Navigate to the \WEB-INF folder for the deployed Web service application. Then navigate to the \WEB-INF\config\serveragent\logging.xml file.

- **J2SE Client Agents** – The logging configuration file for J2SE client agents is found at:

  \oracle\client\owsm\config\interceptors*component_ID*\config\clientagent\logging.xml

  The variable, *component_ID*, is the component ID that was assigned to the client agent when it was registered. After you change the log level, restart the client application.

High-Level Performance Metrics

Policy enforcement points (PEPs) send raw data to the Oracle WSM Monitor, where it is aggregated and compiled to produce high-level indicators pertaining to Web service availability, authentication and authorization results, request latency, and security violations. This information is useful to administrators who want to understand who is accessing the Web services managed by Oracle WSM and how these requests are being handled.

There are two general types of performance metrics:

- Invocation Metrics – Collected each time a service is invoked.
- Flow Metrics – Used to track transactions across an entire business process. Before this type of data can be collected, managed applications must be coded so that they embed flow IDs into all the SOAP message headers they generate.

For high-level performance metrics, your only option is to turn the feature on or off for all metrics. You cannot specify that individual metrics be collected or compiled.

Configuring Agent and Gateway Message Logging

As a feature of policy enforcement, agents and gateways log information from application messages through the use of logging steps specified in agent or gateway enforced policies. See Chapter 5, "Managing Oracle Web Services Manager Policies" for more information on configuring policy steps.

You can configure logging using one or both of the following:

- Log step in the request pipeline that logs information from the service request.
- Log step in the response pipeline that logs information from the service response.

You can add a Log step after any policy step within a pipeline and specify the level of logging for each Log step. Therefore, you can control the amount of information gathered.

You can set the logging function to send log data to one of the following:

- Files located on the same computer as the enforcement component.
- A database repository located on the network.

By default, the output is sent to the Oracle WSM Database. Optionally, you can send output to both the Oracle WSM Database and a local file.

By default, log files are stored in a `logs` directory; each log file name includes the time that the log reached its maximum size and log messages were written to the file.

To meet performance requirements, you should locate the message logs topologically close to the PEP. This implies that a highly distributed environment will require multiple distributed logs.

To send the log entries to a file, edit the cfluent.messagelog.type and cfluent.message.file.logDirectory properties for the policy enforcement point. For more information on editing these properties, see "Making Changes to Your Policy Enforcement Points" on page 9-4.

Changing Maximum Log Entries in Buffer

The Asyncwriter writes log messages to the Oracle WSM Database. By default, the Asyncwriter has a buffer for 1000 log entries. If messages are being sent to the Asyncwriter at a faster rate than the Asyncwriter can write the messages to the database, some of these messages will be lost. This is likely to happen in the case where you are stress testing your Oracle WSM system. If this is the case, then increase the value of the cfluent.messagelog.asynwriter.maxEntries property for the policy enforcement point. See "Making Changes to Your Policy Enforcement Points" on page 9-4 for more information on editing properties for Oracle WSM components.

9

Managing the Oracle Web Services Manager System

This chapter includes the following sections:

Backup and Recovery

For information on backing up and recovering your Oracle Web Services Manager system refer to *Oracle Application Server Administrator's Guide*.

Password Security

A password hash is as sensitive as a password. Never send a password or its hash over an unencrypted communication channel. Oracle recommends that you use HTTPS between all communication points in your Oracle Web Services Manager (Oracle WSM) deployment in order to transmit passwords securely.

Oracle WSM does not support secure JDBC connections.

Changing the Timeout Interval for the Web Browser Session

By default, Web Services Manager Control browser session will time out after 60 minutes. You can change the session timeout by editing the following file:

ORACLE_HOME/owsm/config/ccore/ui-config-common.properties

Edit the ui.session.timeout parameter and enter the number of minutes your browser session will remain active before timing out.

After editing this parameter, use the `wsmadmin deploy control` command and redeploy the Web Services Manager Control application. For more information on redeploying the application, see *Oracle Web Services Manager Deployment Guide*.

If you have installed Oracle WSM as part of Oracle Application Server 10*g* Release 3 (10.1.3.1.0) release, refer to Chapter 14, "OC4J Java Single Sign-On" in *Oracle Containers for J2EE Security Guide* for information on setting your browser session.

Deactivating a Web Service

A service may need to be deactivated for any of the following reasons:

- The service is no longer needed.
- If you change the protocol for a Web service, you must first deactivate the service, then reregister it with the new protocol.
- If you change the service definition or service usage, you must deactivate the service, then reregister it with the new definition.

Once a service is deactivated, you must reregister the service to make it active again. See "To add a Web service to a gateway" on page 2-2 for the procedure to register a Web service.

To deactivate a Web service

1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Register Services.**
2. Click **Services** for the gateway whose service you want to deactivate.
3. Click **Deactivate service** for the service you want to deactivate.

Editing the Web Service Properties

After a service is registered to a gateway, you can return to the service list display and edit service details, for example, changing the WSDL URL for the Web service endpoint, adding Oracle WSM administrators and support groups, updating the service description, or making changes to the service protocol detail information or policies enforced for specific services.

To edit the Web service properties

1. From the navigation pane of Web Services Manager Control, select **Policy Management**, then select **Register Services**.
2. Click **Services** for the gateway whose services you want to view or edit.

   The Web Services Manager Control displays the gateway's current list of services.
3. Click **Edit** for the particular service whose information and details you want to view or update.

   The Web Services Manager Control displays parameters, groups with access, and other information for the selected service.

   From this page, you can update the service description, make changes to the service protocol detail (**Modify Protocol Parameters**), make changes to the policy

enforced for the specific service (**Modify Policy**), or change the Oracle WSM groups who are given Oracle WSM Modify and View permissions for the service.

> **Note:** You can also view and edit information for a specific service policy by selecting **Policy Management**, clicking the **Manage Policies** menu option, selecting a gateway, and then choosing **View Details** or **Edit** for a specific service.

4. When you have finished making changes, click **Save**.

   Oracle WSM saves your changes and displays a confirmation message in Web Services Manager Control.

> **Note:** After making changes to a service registered to an Oracle WSM gateway, **Commit Policy** appears in red in the gateway list display, prompting you to update the gateway with new policy information. Click **Commit** to update the gateway with the service and policy changes you have made.

Publishing Web Services

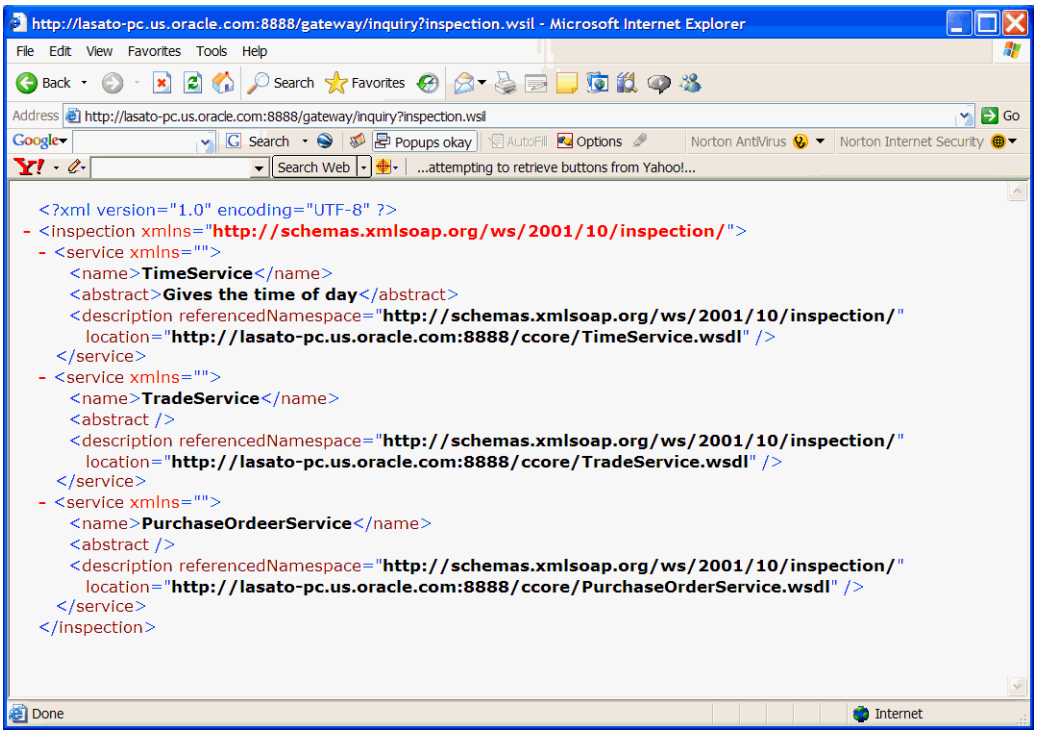
Web services that are registered at a gateway can be made available to others by publishing the following URL:

`http://`*`hostname:port`*`/gateway/inquiry?inspection.wsil`

The variable *hostname* is the Application Server host name and *port* is the port on which the server is listening.

Figure 9–1 shows three services registered at the example gateway.

Figure 9–1 Registered Services at a Gateway



When the URL is entered, the WSIL file is automatically generated, and all Web services registered to the gateway are added to the WSIL file. All information required by WSIL is extracted from the service information.

Making Changes to Your Policy Enforcement Points

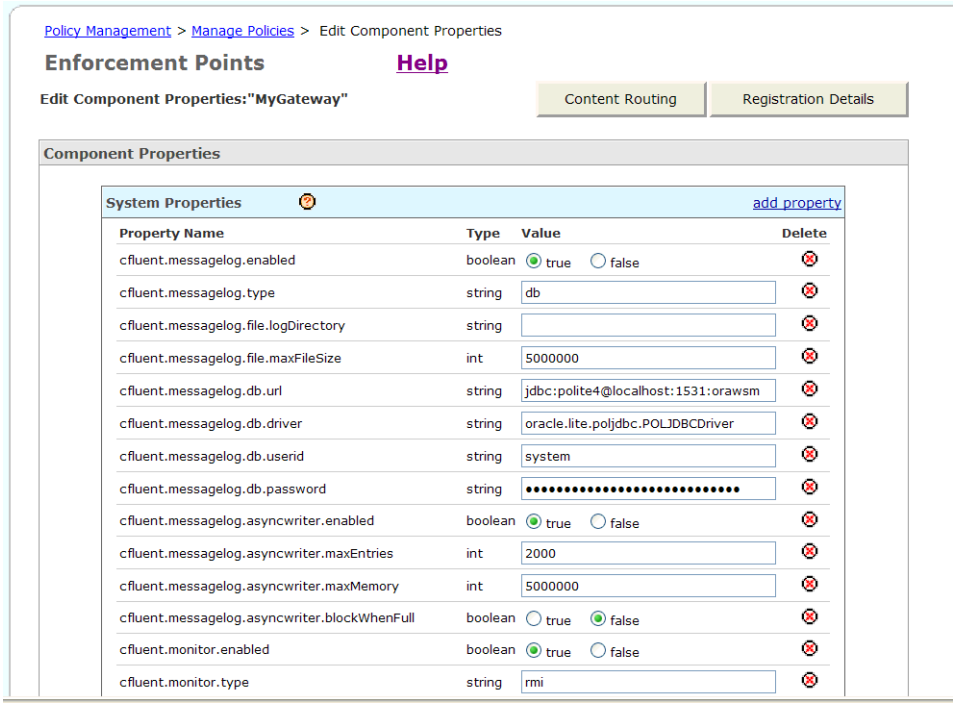
Once a gateway or agent has been registered, you can use Web Services Manager Control and navigate to the Edit Component Properties page, where you can view and make changes to the component. Registered agents and gateways periodically search for changes in the component properties and these changes are communicated to the policy enforcement point.

To navigate to the Edit Component Properties page

1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Manage Policies.**
2. Click the **Edit** icon for the component whose properties you want to edit.

   The Edit Component Properties page is displayed (Figure 9–2).

Figure 9–2 Edit Component Properties Page



3. From the Edit Component Properties page, you can do any of the following:
   - View and edit the properties of the gateway or agent.
   - Click **add property** to add a property for the gateway or agent.
   - Click **Registration Details** to get to the Registration Details page for the gateway or agent. On that page you can view and edit the basic parameters for your component.

To edit the component properties

The properties for the component are displayed on the Edit Component Properties page. Click the question mark (**?**) for help on the properties.

1. From the Edit Component Properties page, edit the desired properties, and click **Save.**

   A confirmation message is displayed.

   Note: The confirmation message reminds you that this change must be committed.

2. Click **Ok.**
3. Commit the changes to have them take effect.

To view and edit the registration details

1. From the Edit Component Properties page, click **Registration Details.**

   The Registration Details page is displayed.

2. Make any changes to the properties on the page, and click **Save.**

A confirmation message is displayed.

3. Click **Ok** to return to the Edit Component Properties page.
4. Click **Save.**

   A confirmation message is displayed.

   Note: The confirmation message reminds you that this change must be committed.

5. Click **Ok.**
6. Commit the changes to have them take effect.

To add a new property

1. From the Edit Component Properties page, click **add property** at the top of the page.
2. Scroll to the bottom of the page where there are several text boxes. Enter the name of the property, select the data type from the list, the default value (optional), and a brief description of the property.
3. Click **Save.**

   A confirmation message is displayed.

   Note: The confirmation message reminds you that this change must be committed.

4. Click **Ok.**
5. Commit the changes to have them take effect.

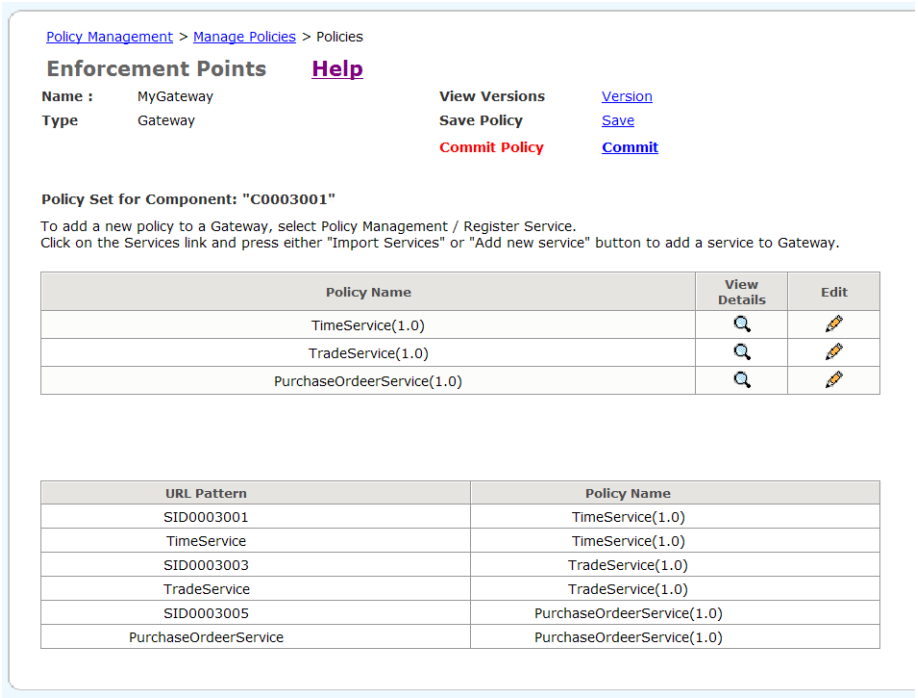
Committing Changes to Policy Enforcement Points

Once a policy enforcement point is registered, you can make changes to the component properties or to its registration details, you can add new properties, and you can edit the policies assigned to the policy enforcement point. In order for any of these changes to take effect, you must commit the changes to the Oracle WSM Database.

To commit your changes

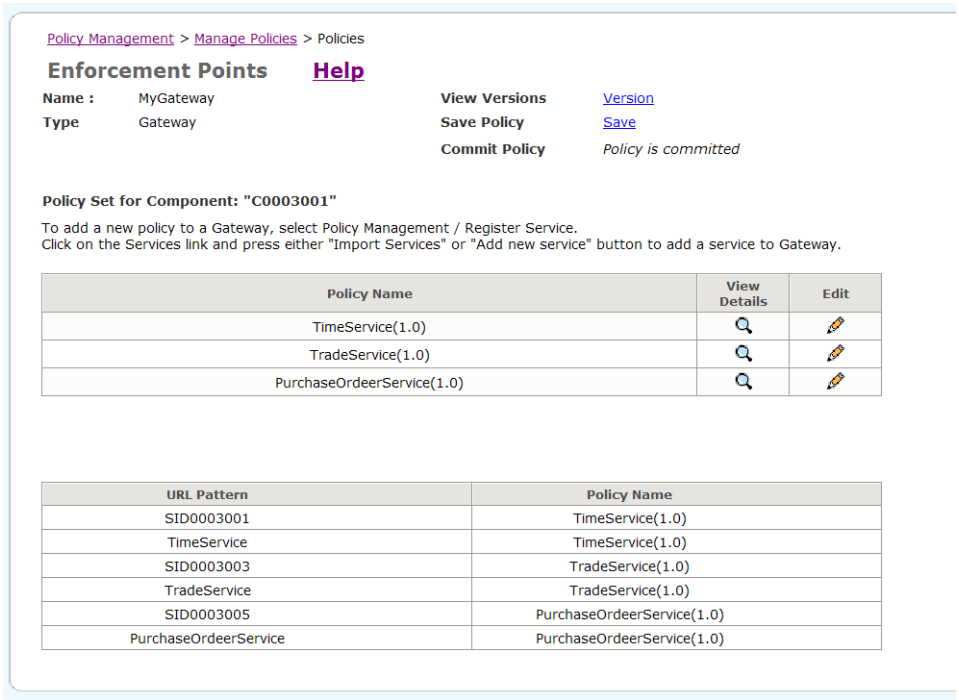
1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Manage Policies.**
2. In the List of Components, click **Policies** for the desired gateway or agent.
3. In the Policies page, if the Commit Policy field appears in red text, click **Commit** to commit any outstanding changes that have been made to the policy enforcement point.

Figure 9–3 Commit Policy Field in Red Text



The page refreshes and the commit link is replaced with the message *Policy is committed.*

Figure 9–4 Policies Page Showing the Policy Is Committed



Configuring Connection Time Out for Authentication Sources

You can configure when connections to the LDAP directory and Active Directory server time out. By default, the connections do not time out.

Each instance of a policy step creates one or two long-lived connections to LDAP directory or Active Directory. If your system is configured to forcibly terminate long-lived connections, you may want to configure a connection time out to avoid request failures. Set the time out value to be close to, but lower than the time out value used for the enforced terminations.

To configure the connection time out

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 connection lifetime parameter under java-options. For example:

   `-Doracle.wsm.directory.timeout=3600000`

   The value indicates the number of milliseconds Oracle Web Services Manager will try to connect to the LDAP server or Active Directory Server before it times out.

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

Configuring the Number of Messages Displayed

When you display the message logs for a component, by default, Oracle WSM displays up to 500 of the most recent message logs. If there are more than 500 message logs, you will not be able to view messages 501 and beyond. You can configure the number of messages that are displayed by editing the following file:

ORACLE_HOME/owsm/config/ccore/ui-config-installer.properties

Edit the ui.messagelog.maxViewableMessageLogs property and specify the maximum number of messages you want displayed. By default, this property is set to 500.

ui.messagelog.maxViewableMessageLogs=500

If you do not want to set a limit, that is, you want to be able to see all messages, then specify zero (0), as in

ui.messagelog.maxViewableMessageLogs=0

Purging Message Logs

Using Web Services Manager Control, you can delete message logs that you no longer need from Oracle WSM Database.

To purge message logs

1. From the navigation pane of Web Services Manager Control, select **Operational Management,** then select **Overall Statistics.** Select **Message Logs,** then select **Purge Message Logs.**
2. In the Component list, select the component whose message logs you want to delete.
3. In the Time Range list, select the period for which you want to delete the message logs.

   Note: For example, if you select *Before 60 days,* any message logs with timestamps earlier than the previous 60 days will be deleted. If the current date and time is March 31, 2007, 8:00 a.m., then message logs with timestamps before January 30, 2007, 8:00 a.m., will be deleted. Message logs with timestamps from January 30, 2007, 8:00 a.m. through March 31, 2007, 8:00 a.m. are preserved.

4. A message appears asking you to confirm that you want to purge the message logs. Click **OK.**

   A message confirms that the purge was successful.

Creating Indexes Against the PIPELINES Table

Oracle has provided a SQL file that you can execute to create the following indexes on the PIPELINES table:

- IDX_PIPELINES_POLICY_ID index on the POLICY_ID column
- IDX_PIPELINES_MAJOR_MINOR_VER index on the PIPELINE_MAJOR_VER and PIPELINE_MINOR_VER columns

Creating Indexes for the Oracle Database

Complete the following steps to create the indexes for the Oracle Database and Oracle Lite Database.

1. Locate the SQL file used to generate the indexes for your database in the following location:

   ORACLE_HOME/config/db/oracle/PolicyRepository/PipelinesIndexes.sql

   The variable, *ORACLE_HOME*, is the location where Oracle WSM is installed.
2. Copy the SQL file to the *ORACLE_HOME* where Oracle Database is installed.
3. Connect to the Oracle Database and execute the SQL script.

Creating Indexes for the Oracle Lite Database

Complete the following steps to create the indexes for the Oracle Lite Database.

1. Locate the SQL file used to generate the indexes for your database in the following location:

ORACLE_HOME/config/db/olite/PolicyRepository/PipelinesIndexes.sql

The variable, *ORACLE_HOME*, is the location where Oracle WSM is installed.

2. Copy the SQL file to the *ORACLE_HOME* where Oracle Lite is installed.
3. Connect to the Oracle Lite Database and execute the SQL script.

Changing the HTTP Port on Oracle Application Server

If you change the port on which Oracle Application Server listens for HTTP requests, then the ports on which the Oracle WSM Policy Manager and Web Services Manager Control are listening for HTTP will also change. Therefore, other applications that communicate with Oracle WSM Policy Manager and Web Services Manager Control must be updated with the new HTTP port. For more information on the interaction between Oracle WSM components, see the Oracle WSM deployment diagram in Chapter 1, "Planning an Oracle Web Services Manager Deployment" in *Oracle Web Services Manager Deployment Guide*.

The following procedures must be performed to make the required changes.

Oracle WSM Policy Manager

Web Services Manager Control, Oracle WSM Gateway, and Oracle WSM Agents communicate with Oracle WSM Policy Manager. Therefore, you must reconfigure these applications to point to the new Oracle WSM Policy Manager endpoint.

To reconfigure Web Services Manager Control

1. Edit the following files:
   - *ORACLE_HOME*/owsm/config/ccore/ui-config-installer.properties
   - *ORACLE_HOME*/owsm/config/ccore/policyui-config-installer.properties

   In each file, edit the `ui.pm.server.httpPort` property with the new HTTP port.
2. Redeploy the Web Services Manager Control using the `wsmadmin deploy` command.

   See *Oracle Web Services Manager Deployment Guide* for more information on the `wsmadmin deploy` command.

To reconfigure Oracle WSM Gateway

1. In the *ORACLE_HOME*/owsm/config/ccore/gateway-config-installer.properties file, edit the gateway.policymanagerURL property with the updated URL to the Oracle WSM Policy Manager.
2. Redeploy the Oracle WSM Gateway using the `wsmadmin deploy` command.

   See *Oracle Web Services Manager Deployment Guide* for more information on the `wsmadmin deploy` command.

To reconfigure Oracle WSM Server Agents

1. In the *ORACLE_HOME*/owsm/config/serveragent/serveragent-config-installer.properties file, edit the agent.policymanagerURL property with the updated URL to the Oracle WSM Policy Manager.
2. Redeploy the server agent using the `wsmadmin deploy` command.

See *Oracle Web Services Manager Deployment Guide* for more information on the `wsmadmin deploy` command.

To reconfigure Oracle WSM Client Agents

1. In the *ORACLE_HOME*/owsm/config/clientagent/clientagent-config-installer.properties file, edit the agent.policymanagerURL property with the updated URL to the Oracle WSM Policy Manager.
2. Redeploy the client agent using the `wsmadmin deploy` command.

   See *Oracle Web Services Manager Deployment Guide* for more information on the `wsmadmin deploy` command.

Note: For OC4J Server Agents and J2EE client agents, you can avoid redeploying the agent by updating the following files:

- *ORACLE_HOME*/owsm/config/interceptors/*component_ID*/config/clientagent/clientagent-config-installer.properties
- *ORACLE_HOME*/owsm/config/interceptors/*component_ID*/config/serveragent/serveragent-config-installer.properties

The variable, *component_ID*, is the component ID of the agent.

Edit the agent.policymanagerURL property with the updated URL to the Oracle WSM Policy Manager.

Oracle WSM Gateway

When the HTTP port changes, the virtualized endpoints that Oracle WSM Gateway exposes to Web services clients also change. The URL for the Oracle WSM Gateway must be updated, and any Web services that use that virtualized URL must also be updated.

To update the Oracle WSM Gateway URL

1. From Web Services Manager Control, navigate to the Policy Management page.
2. Click **Edit** for the Oracle WSM Gateway.
3. Click **Registration Details.**
4. Edit the Component URL property with the updated URL.

Web Services Clients

Web services clients that use the virtualized Web service URL and WSDL exposed by the Oracle WSM Gateway must be reconfigured and redeployed with the new Web service URL and WSDL values.

10

Troubleshooting

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

Limitations of Java Policy Enforcement Points

Policy enforcement points (PEP) that rely on Java resources (that is, gateways and most agents) have the following limitations:

- Java PEPs do not support actors or roles inside SOAP messages. Java PEPs usually process security headers, even if those security headers include actor or role information, which ought to prevent such processing.
- If a SOAP message (request) contains one security header that does not contain an actor, the Java PEP often adds a second security header, which also does not contain any actor information. This action violates the WS-Security specification.

SSL Does Not Work Properly on OC4J

If you are using Secure Sockets Layer (SSL) libraries, remove the jssl-1_1.jar file from the OC4J_Home/lib directory, and replace it with the jssl-1_2.jar file. If you are using some other SSL library, simply remove the jssl-1_1.jar file from the OC4J_Home/lib directory. Once you have completed this step, you can use any other vendor's implementation of Java Secure Socket Extension (JSSE).

Report Engine Does Not Display Properly

To enable the Report engine, verify that all of the following are true:

- You have met the requirements listed in the sections on required X Window System (X11) packages for Linux and Solaris and the required DISPLAY variable for Linux and Solaris in *Oracle Application Server Installation Guide* for your platform.
- You can verify that X-client is running on the computer pointed to by the DISPLAY variable for the computer running Corda.

Make certain that the Corda chart server is running. Solaris systems require X Server to display Operational Management charts and graphs. If X Server is not installed, the images will not display properly. For more information on X Server, refer to the following URL:

`http://developers.sun.com/solaris/articles/solaris_graphics.html`

Error When Importing WSIL

Problem

You are registering a service to a gateway by importing a Web service from a UDDI or a WSIL; you get an error similar to one of the following:

- Could not establish a connection to the URL: *URL_address*
- Error accessing the specified URL

Solution

The cause of this problem may be that your proxy server settings are not set. Follow the procedure that follows and set your proxy server settings.

For standalone Oracle WSM

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 = *host_name*

   Table 10–1 describes how to set the values for the parameters.

Table 10–1 Parameter Settings for a Proxy Server

Parameter Value	Description of Value
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.
host_name	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

3. Restart the Oracle WSM server for the configuration changes to take effect.

For Oracle WSM when it is installed as part of Oracle Application Server 10g Release 3 (10.1.3.1.0)

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 = host_name
```

Table 10–2 Parameter Settings for Oracle WSM Installed as Part of Oracle Application Server

Parameter Value	Description of Value
`true/false`	The value `true` enables the proxy server.
`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
`host_name`	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`

5. Restart the server for the configuration changes to take effect.

Example UDDI Registry Does Not Work

Problem

When you try to import a service using the example UDDI registry `http://<oc4jhost>:<port>/registry/uddi/inquiry`, you get the following error: Could not establish a connection to the URL: http://*oc4jhost:port*/registry/uddi/inquiry.

Solution

You must install the Oracle Registry in order to use this UDDI.

Error When Accessing WSDL

Problem

When you enter or paste the Web service WSDL URL in a browser, you get the following error:

Oracle Web Services Manager Gateway C0003001: No policies found for service "SID0003003". Make sure the service is registered correctly and the gateway policies are up-to-date.

Solution

One possible cause is that the component ID for the gateway is incorrectly set. Follow this procedure:

1. From the navigation pane, select **Policy Management,** then select **Manage Policies.**
2. Look for your gateway in the list of components, and get the component ID.
3. Open the following file:

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

   Verify that the gateway.component.ID property is set to the same ID as the ID for your gateway. If they are not the same, then make the necessary change to the file.

If you change the component ID in the file, you must redeploy the gateway application for the changes to take effect. See *Oracle Web Services Manager Deployment Guide* for more information on deploying your application.

Error Testing Access to Web Service

Problem

You are using the Test Page to test access to your Web service. You enter the WSDL URL and click **Submit Query**, and get the following error: Failed to read WSDL from http://*host_name*:*port*/gateway/services/SID000300*n*?wsdl:WSDL not found.

Solution

- One possible cause is that the component ID for the gateway is incorrectly set. Follow the procedure in "Error When Accessing WSDL" on page 10-3.
- Another possible cause is that the proxy server settings are not set. See the solution for "Error When Importing WSIL" on page 10-2.

Cannot Access Policy Management Menu

Problem

You are running standalone Oracle WSM. You log in to Web Services Manager Control, and you get the error: 500 Server Error. You are able to click on and access the Operational Management, Tools, and Administration menus. You see this error only when trying to access the Policy Management menu.

Solution

The problem is caused because the nonproxy hosts settings were not set during installation. See the solution for "Error When Importing WSIL" on page 10-2.

Web Services Manager Control Times Out

Problem

The Web Services Manager Control times out too quickly, and you have to log in to your session.

Solution

By default, the browser session is set to 60 minutes. Edit the ui.session.timeout parameter in the *ORACLE_HOME*/owsm/config/ccore/ui-config-common.properties file.

Metrics Data Does Not Appear in Web Services Manager Control

Problem

You see the metrics for a Web service invocation for a period of time. After some time, the metrics for that particular invocation stop appearing in Web Services Manager Control.

Solution

You can view metrics for Oracle Web Services Manager for different time periods ranging from the last 10 minutes up to 60 days. However, by default, Oracle WSM Database persists data only for the last 100 minutes. Therefore, to see historical data beyond 100 minutes, you must configure the monitor.aggregator.measurementStore.WindowSize parameter in the *ORACLE_HOME*/owsm/config/coreman/monitor-config-installer.properties file. For more information, see "Configuring Metrics Data Persistence" on page 6-28.

Log Files Providing Wrong Level of Information

Problem

Log files do not contain the right level of information. More or less information is needed.

Solution

Each Oracle Web Services Manager application has its own logging configuration file.

For more information on changing the logging level, see "Low-Level Event and State Logs" on page 8-1.

Resetting Log Levels Does Not Seem to Work

Problem

Changing the log level does not change the level of information in the log files.

Solution

- Each application, Oracle WSM Policy Manager, Oracle WSM Monitor, Oracle WSM Gateway, Oracle WSM Server Agent, Oracle WSM Client, and Web Services Manager Control, has its own logging configuration file. Verify that you have configured the correct file.
- You can set the log level for each class. In most instances, you will probably set the same log level for all classes for an application. But if you have set different log levels, verify that the log level is set as you desire for the correct class.
- After you configure the log level, you must redeploy the application for the changes to take effect.

Addressing Performance Issues

A comprehensive treatment of performance issues within distributed systems involves a level of complexity that is beyond the scope of this book. There are many good reference guides that present guidelines, in addition to details about performance issues and performance tuning.

As an alternative to providing a comprehensive discussion of performance, this section offers some general guidelines on how to identify a performance bottleneck and how to approach addressing such problems.

If you discover a performance bottleneck, you should first check to see that you have addressed the expected traffic load throughout your Web services deployment. If there is a system in the critical path that is at 100% CPU usage, you may simply need to add one or more computers to the cluster.

If there is a bottleneck in your deployment, it is likely to be within one of the following:

- Traffic through a slow connection with an agent
- Traffic through a slow connection with a gateway
- Unexpected high traffic volume through an Oracle WSM Monitor that is connected to a database
- Latency in connections to third-party queueing systems like JMS or MQ

For any of these problems, check the following potential sources:

- Problems with policy steps that include connections to outside resources, especially the following types:
  - Database repositories
  - LDAP repositories
  - Secured resources
  - Proprietary security systems
- Problems with database performance

If you identify one of these as the cause of a bottleneck, you may need to change how you are handling your database or LDAP connections, the securing of your resources, or the details of how you are aggregating and persisting Oracle WSM monitoring data.

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.

A

Oracle Web Services Manager Policy Steps

This appendix is a reference for the Oracle Web Services Manager (Oracle WSM) policy steps.

Table A–1 shows which policy steps can be used with each policy enforcement point.

Note: Oracle Web Services Manager supports SOAP version 1.1 only.

Table A–1 Supported Policy Steps for Policy Enforcement Points [1]

Steps	Gateways	Agents[2]			
		OC4J[3]		AXIS[4]	
		Client	Server	Client	Server
Active Directory Authenticate	X	X	X	X	X
Active Directory Authorize	X	X	X	X	X
Decrypt and Verify Signature	X	X	X	X	X
Extract Credentials	X	X	X	X	X
File Authenticate	X	X	X	X	X
File Authorize	X	X	X	X	X
Handle Generic Fault	X	NA	NA	X	X
Insert Oracle Access Manager Token	X	NA	NA	X	X
Insert WSBASIC Credentials	X	X	X	X	X
LDAP Authenticate	X	X	X	X	X
LDAP Authorize	X	X	X	X	X
Log	X	X	X	X	X
Oracle Access Manager Authenticate Authorize	X	X	X	X	X
SAML – Insert WSS 1.0 Sender-Vouches Token	X	NA	NA	X	NA
SAML – Verify WSS 1.0 Token	X	X	X	NA	X
Sign Message	X	X	X	X	X
Sign Message and Encrypt	X	X	X	X	X

Table A–1 (Cont.) Supported Policy Steps for Policy Enforcement Points [1]

Steps	Gateways	Agents[2]			
		OC4J[3]		AXIS[4]	
		Client	Server	Client	Server
SiteMinder Authenticate	X	X	X	X	X
SiteMinder Authorize	X	X	X	X	X
Verify Certificate	X	X	X	X	X
Verify Signature	X	X	X	X	X
XML Decrypt	X	X	X	X	X
XML Encrypt	X	X	X	X	X
XML Transform	X	X	X	X	X

[1] NA = Not Applicable

[2] For more information on the different types of agents, see *Oracle Web Services Manager Deployment Guide*.

[3] OC4J agents are native to OC4J. Using this type of agent requires that you have OC4J administrator permissions to deploy the agent.

[4] AXIS agents are filter agents and are injected into the Web service or client application. AXIS agents are used to protect AXIS stack-based Web services.

Active Directory Authenticate

Verifies the sender's identity using Microsoft Active Directory.

Usage

Uses a user name and password to authenticate the sender.

Prerequisite Steps

Extract Credentials

Properties

Table A–2 Active Directory Properties

Property	Description
Enabled	If set to true, this step is enabled.
AD host	Host name on which the Active Directory server is running that contains the user schema.
AD port	Port on which the Active Directory server is listening for the connections.
AD SSL port	Port on which the Active Directory server is listening for SSL connections.
AD baseDN	Base distinguished name where the users and groups (also known as roles) data exist for this Active Directory server.
AD domain	Active Directory domain of the user. In the example john.doe@oracle.com, the domain *oracle.com* would be specified.
ADSSLEnabled	If set to true, then the connection to Active Directory uses SSL.
Uid Attribute	Attribute that uniquely identifies the user. This is used in the search filter.
User Attributes to be retrieved	User profile attributes to be read after authentication. These attributes can be used in subsequent steps such as SAML - Insert WSS 1.0 Sender-Vouches Token, which inserts attribute statements using the retrieved values. Custom policy steps can also use these attributes.

Possible Next Steps

Active Directory Authorize

Active Directory Authorize

Grants or denies the sender's request using Microsoft Active Directory.

Usage

Authorizes access to the service based on user group membership in Active Directory. The user must be a member of one of the configured groups in the ServiceRoles property to be granted access.

Prerequisite Steps

Active Directory Authenticate

Properties

Table A–3 Active Directory Authorize Properties

Property	Description
Enabled	If set to true, this step is enabled.
AD host	Host name on which the Active Directory server is running that contains the users and their roles.
AD port	Port on which the Active Directory server is listening for the connections.
AD SSL port	Port on which the Active Directory server is listening for SSL connections.
AD baseDN	Base distinguished name where the users and groups (also known as roles) data exist for this Active Directory server.
ServiceRoles	Comma-delimited list of service roles that have access to the service.
ADAdminUser	Admin user with permission to connect to the Active Directory server and perform searches on the schema.
ADAdminPwd	Password for the Admin user with permission to connect to the Active Directory server.
AD domain	Active Directory domain of the user. In the example john.doe@oracle.com, the domain *oracle.com* would be specified.
ADSSLEnabled	Set this to true if the Active Directory connection must be an SSL connection.
Uid Attribute	An attribute, such as *uid*, that uniquely identifies the user entry in Active Directory.

Possible Next Steps

There are no recommended next steps.

Decrypt and Verify Signature

Decrypts the XML message and verifies that the signature is valid.

Usage

Decrypts the message, then verifies the signature. You can use this policy step only if the order in which the message was secured was by being signed first, and then encrypted.

For all other situations, use the individual policy steps, Verify Signature and XML Decrypt. For example:

- If the encryption and signing were done in the opposite order, that is, the message was encrypted before it was signed, then use the Verify Signature step followed by the XML Decrypt step. This is true whether the encryption and signing were done in a single step or as separate steps.
- If the message was encrypted and not signed, then use the Decrypt XML step.
- If the message was signed and not encrypted, then use the Verify Signature step.

Prerequisite Steps

None

Properties

Table A–4 Decrypt and Verify Signature Properties

Property	Description
Enabled	If set to true, this step is enabled.
Decryptor's keystore location	Location of the keystore on the local file system that contains the private keys used for decryption.
Decrypt Keystore Type	Keystore file format. The valid values are: ■ jks – Java keystore format ■ PKCS12 – Public Key Cryptographic Standard #12 format
Decryptor's keystore password	Password to access the decryptor's keystore.
Decryptor's private-key alias	Alias of the private key used for decryption.
Enforce Encryption	If set to true, Oracle WSM does not allow an unencrypted message to pass through.
Verifying Keystore location	Location of the keystore on the local file system that contains the public key used for signature verification.
Verifying Keystore type	Keystore file format. The valid values are: ■ jks – Java keystore format ■ PKCS12 – Public Key Cryptographic Standard #12 format
Verifying Keystore password	Password to access the verifying keystore.
Signer's public-key alias	Alias of the public key used for signature verification.
Remove Signatures	If selected, the signature is removed from the SOAP security header after successful verification.

Table A–4 (Cont.) Decrypt and Verify Signature Properties

Property	Description
Enforce Signing	If set to true, Oracle WSM does not allow an unsigned message to pass through.

Possible Next Steps

Extract Credentials

Extract Credentials

Locates and extracts credentials and presents the credentials in a form that can be authenticated. You must know from where the credentials are to be extracted.

Prerequisite Steps

If the message was protected, then the appropriate steps required to decrypt the XML message or verify the signature, or do both, must first be performed.

Properties

Table A–5 Extract Credentials Properties

Property	Description
Enabled	If set to true, this property is enabled.
Credentials location	Where the credentials are extracted. The four possible locations are: ■ HTTP `Authorization` header – Specify *HTTP*. This is the default. Authorization is provided using the HTTP basic authorization scheme (BASIC-AUTH). ■ WS-BASIC SOAP security header – Specify *WS-BASIC*. Credentials are extracted from the standard UsernameToken as specified in the WS-I Basic Security Profile. Only plain text passwords are supported. ■ XPath – Specify the XPath expression to the credentials. Do not enter the word *XPath*. Start with the slash (/). For example: `/soap:Header/soap:Envelope/wsse:Security/wsse:UsernameToken/` XPath expressions are used to extract the user name and password from anywhere in the SOAP envelope. You must specify additional properties (Namespaces, UserID xpath, and Password xpath).
Namespaces	Comma-delimited list of prefix and namespace Uniform Resource Identifier (**URI**) pairs for the prefixes used in the User ID xpath and Password xpath properties. For example: `soap=http://schemas.xmlsoap.org/soap/envelope, wsse=http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd` This parameter applies only if the *Credentials location* property is specified with an XPath expression.
UserID xpath	XPath for the user name. This XPath is relative to the XPath specified in the *Credentials location* property. For example: `wsse:Username` This parameter only applies if the *Credentials location* property is specified with an XPath expression.
Password xpath	XPath for the password. This XPath is relative to the XPath specified in the *Credentials location* property. For example: `wsse:Password` This parameter applies only if the *Credentials location* property is specified with an XPath expression.

Possible Next Steps

The next step is to authenticate the credentials using one of the following steps: Active Directory Authenticate, Oracle Access Manager Authenticate Authorize, File Authenticate, LDAP Authenticate, or SiteMinder Authentication.

File Authenticate

Verifies the sender's identity by checking against entries in a file.

Usage

Used most often in testing situations. The file format is the same as the .htpasswd file format used by the Apache Web server. The password can be encoded in four forms: MD5, SHA1, plain text, or some mix of the three forms.

The MD5 format used by Oracle Web Services Manager is not compatible with other MD5 encodings. Therefore, if you use the MD5 encoding, you must use the tool provided to encode the passwords.

The `wsmadmin` command-line tool can be found at the following location:

ORACLE_HOME`/OWSM_1/owsm/bin`

Create a text file with the user name and password in unencrypted text. For example, the text file, password.txt, could contain the following entries:

```
johndoe:baseball
janedoe:rollarskating
```

You must run the `md5encode` command separately for each user name and password combination.

The command to run the tool is:

`wsmadmin md5encode` *`htpasswdfile user_name password`*

The parameters are:

- *`htpasswdfile`* – the name of the file to which the user name and password are added
- *`user_name`* – the user name in the text file
- *`password`* – Password assigned to the user

For example:

ORACLE_HOME`/OWSM_1/owsm/bin/wsdadmin.sh md5encode johndoe baseball C:/password.txt`

The `wsmadmin` tool encrypts the password and replaces the password you entered in unencrypted text with the encrypted form.The following are example entries in the file after the command has been executed:

```
johndoe:{MD5}JMnhX1KvxHwiW3V+e+4fnQ==
janedoe:{MD5}dqIXO+Y5M1TnL/pNbfEDCg==
```

Prerequisite Steps

Extract Credentials

Properties

Table A–6 File Authenticate Properties

Property	Description
Enabled	If set to true, this step is enabled.

Table A–6 (Cont.) File Authenticate Properties

Property	Description
Passwd file location	Location of the file that contains the user names and passwords. You can provide a full path or a relative path. For the gateway and OC4J agents, the path is relative to *ORACLE_HOME/j2ee/home*.
.htpasswd file format	Format in which the passwords are encrypted. The valid values are: ■ md5 – Passwords encrypted using Message-Digest Algorithm 5 (MD5). ■ sha – Passwords encrypted using Secure Hash Algorithm (SHA). ■ plaintext – Unencrypted passwords in plain text. ■ mixed – Passwords using a combination of one or more of the supported formats. This is the default.

Possible Next Steps

File Authorize

File Authorize

Grants or denies access to an authenticated user using a local roles file.

Usage

Used most often in testing situations.

Role information is defined in a text file with the following format:

```
<user username="name_of_user" roles=role_1, role_2, role_n"/>
```

Each entry identifies the user and the roles to which the user is assigned. The entry for each user is on a separate line in the file. An example file can be found in the following location: *ORACLE_HOME*/owsm/config/gateway/roles.xml.

If any of the roles to which the user is assigned matches one of the roles defined in the Allowed Roles property, the user is granted access to the service.

Prerequisite Steps

File Authenticate

Properties

Table A–7 File Authorize Properties

Property	Description
Enabled	If set to true, this step is enabled.
User roles file location	Location of the file describing the user roles. You can provide a full path or a relative path. For the gateway and OC4J agents, the path is relative to *ORACLE_HOME/j2ee/home*.
Allowed roles	Comma-delimited list of roles authorized access to the service.

Possible Next Steps

There are no recommended next steps.

Handle Generic Fault

Provides custom message in the SOAP fault when errors are encountered.

Usage

Customizes the message that is sent back in the SOAP fault when errors occur in processing the policy.

Prerequisite Steps

None

Properties

Table A–8 Handle Generic Fault Properties

Property	Description
Enabled	If set to true, this step is enabled.
CustomMessage	Message text that the error handler returns. This message overrides the default Oracle WSM error message.

Possible Next Steps

There are no recommended next steps.

Insert Oracle Access Manager Token

Inserts an ObSSOCookie in the SOAP security header.

Usage

Used with the gateway policy enforcement points when the client sends an ObSSOCookie in the HTTP header, and the Web service expects the ObSSOCookie in a SOAP security header.

Prerequisite Steps

None

Properties

Table A–9 Insert Oracle Access Manager Token Properties

Property	Description
Enabled	If set to true, this step is enabled.

Possible Next Steps

To protect the token, use the Sign Message or the Sign Message and Encrypt policy step.

Insert WSBASIC Credentials

Inserts user name and password credentials in a SOAP security header.

Usage

Used with gateway policy enforcement points when the client credentials are specified in one format and the Web service expects the credentials in a WS-BASIC SOAP header. You must first use the Extract Credentials step to get the credentials, then use Insert WSBASIC Credentials to put the credentials in the SOAP header as specified in the Web Services Security Username Token Profile 1.0.

Prerequisite Steps

Extract Credentials

Properties

Table A–10 Insert WSBASIC Credentials Properties

Property	Description
Enabled	If set to true, this step is enabled.
User Name	User name for the user's credentials.
User Password	Password for the user's credentials.

Possible Next Steps

To protect the token, this step should be followed by the Sign Message and Encrypt policy step.

LDAP Authenticate

Verifies the sender's identity by checking the user name and password in an LDAP directory.

Usage

Establishes that a valid client is invoking the Web service.

Prerequisite Steps

Extract Credentials

Properties

Table A–11 LDAP Authenticate Properties

Property	Description
Enabled	If set to true, this step is enabled.
LDAP host	Host on which the LDAP directory server is running that contains the user schema that must be authenticated.
LDAP port	Port on which the LDAP directory server is listening for the connections.
LDAP SSL port	Port on which the LDAP directory server is listening for SSL connections.
User objectclass	The object class of the user for which authentication is being performed.
LDAP baseDN	The base distinguished name where the users and groups (also known as roles) data exist for this LDAP directory server.
LDAP adminDN	This property is required when the *LDAP admin login enabled* property is set to true. The distinguished name for Admin. For example, *cn=DirectoryManager.*
LDAP admin password	The password for the Admin user when the *LDAP admin login enabled* property is set to true.
LDAP admin login enabled	Set to true if the Admin user is required to connect to the LDAP directory server. If set to false, then anonymous access is permitted.
LDAPSSLEnabled	Set to true if the LDAP connection must be an SSL connection.
Uid Attribute	A property, such as *uid*, that uniquely identifies the user entry in LDAP.
User Attributes to be retrieved	User profile properties to be read. These properties can be used in subsequent steps such as SAML - Insert WSS 1.0 Sender-Vouches Token which inserts attribute statements using the retrieved values. Custom policy steps can also use these attributes.

Possible Next Steps

LDAP Authorize

LDAP Authorize

Grants or denies access to an authenticated user using an LDAP directory server.

Usage

Authorizes access to the Web service based on user group membership in LDAP. The user must be a member of one of the configured groups in the ServiceRoles property to be granted access.

Prerequisite Steps

LDAP Authenticate

Properties

Table A–12 LDAP Authorize Properties

Property	Description
Enabled	If set to true, this step is enabled.
LDAP host	Host on which the LDAP directory server is running that contains the users and their roles.
LDAP port	Port on which the LDAP directory server is listening for connections.
LDAP SSL port	Port on which the LDAP directory server is listening for SSL connections.
LDAP baseDN	The base distinguished name where the users and groups (also known as roles) data exist for this LDAP directory server. Set the base distinguished name to the root DN of both the users and groups. For example, if the users are in cn=users, dc=company, dc=com, and the groups are in cn=groups, dc=company, dc=com, then the LDAP baseDN should be set to dc=company, dc=com.
ServiceRoles	Roles that have access to the service. Use an asterisk to indicate white spaces in the name. For example, for the role *Customer Support*, you would specify *Customer*Support*.
LDAPAdminDN	Distinguished name for Admin. This property is required when *LDAPAdminLoginEnabled* property is set to true. If set to false, then an anonymous bind is permitted.
LDAPAdminPwd	The password for the Admin user when the LDAPAdminLoginEnabled property is set to true.
LDAPAdminLoginEnabled	Set to true if Admin user is required to connect to the LDAP directory server. If set to false, then an anonymous bind is permitted.
LDAPSSLEnabled	Set to true if the connection to LDAP must use SSL.
Uid Attribute	An attribute, such as *uid*, that uniquely identifies the user entry in LDAP.
LDAP Group Object Class	Name of objectclass for LDAP groups.

Possible Next Steps

There are no recommended next steps.

Log

Logs the current message as received in this policy step.

Usage

Debugs other policy steps. Insert it after the policy step you want to debug to find out how the message was modified by the step.

Messages are stored in the database and can be viewed from Web Services Manager Control.

Prerequisite Steps

None

Properties

Table A–13 Log Properties

Property	Description
Enabled	If set to true, this step is enabled.
Log level	The part of the message you want logged. Valid values are: envelope, body, header, and all. Note, the values *envelope* and *all* are identical.

Possible Next Steps

There are no recommended policy steps.

Oracle Access Manager Authenticate Authorize

Verifies the sender's identity in Oracle Access Manager, and if the sender is authenticated, the sender is given access using an Oracle Access Manager policy.

Usage

Combines authentication and authorization in a single step. Authentication is automatically performed, and authorization is performed, by default. You may turn authorization off with the Authorize parameter.

Oracle Access Manager extracts the credentials to authenticate the sender in the order shown:

1. ObSSOCookie in the HTTP header
2. ObSSOCookie in SOAP header
3. User name and password extracted in a previous Extract Credentials step
4. Signing certificate extracted in a previous Verify Signature step
5. SSL certificate used by the transport layer

The ObSSOCookie in the SOAP header must be set as a `BinarySecurityToken` with `Security/BinarySecurityToken/@ValueType = ObSSOCookie.`

Oracle Access Manager first checks for an ObSSOCookie in the HTTP header. If it finds the cookie, it uses this to authenticate the sender. If not, it continues and checks for an ObSSOCookie in the SOAP header. It continues searching in the order previously shown until it finds the credential.

If the Authorize property is set to true, then authorization is performed based on a policy that is configured in Oracle Access Manager with an authorization rule. The following types of authorization can be performed:

- Membership in a group (static or dynamic)
- Time-of-day based authorization
- Internet Protocol (IP) validation
- Custom authorization using Oracle Access Manager authorization plug-ins

Required Configuration

Oracle Access Manager Authenticate Authorize uses the Java Native Interface (JNI) libraries. Therefore, you must set up your environment variables to load the shared libraries. See Appendix E, "Authentication Sources," in *Oracle Web Services Manager Deployment Guide* for more information.

User Attribute or Group Information

You may need to retrieve user attributes or group information if a subsequent policy step, such as SAML – Insert WSS 1.0 Sender-Vouches Token or a custom policy step, requires this information. This information can be retrieved using the Oracle Access Manager Authenticate Authorize step. You must use the return action functionality in Oracle Access Manager. Use the Access System Console to add an action for an authentication rule. On the Actions page, find the Authentication Success section, then find the Return section. Specify values for the following fields.

To retrieve user attributes, enter the following values:

- **Type –** `HeaderVar`

- **Name –** Enter any name of your choice to identify the attribute. Note, however, this is the name by which the attribute *must* be referred to in subsequent policy steps. For example, this is the name you would enter in the *User Attributes for attribute statements* parameter for the SAML – Insert WSS 1.0 Sender-Vouches Token policy step.
- **Return Attribute –** Enter the LDAP user attribute name. You must use the name by which the attribute is identified in the LDAP directory.

To retrieve user groups, enter the following values:

- **Type –** `HeaderVar`
- **Name –** Enter any name of your choice to identify the attribute. Note, however, this is the name by which the attribute *must* be referred to in subsequent policy steps. For example, this is the name you would enter in the *User Attributes for attribute statement*s parameter for the SAML – Insert WSS 1.0 Sender-Vouches Token policy step or in a custom step.
- **Return Attribute –** `obmygroups`

Retrieved attribute or group information can be used in the SAML – Insert WSS 1.0 Sender-Vouches Token policy step or in a custom step.

For more information on how to set up return actions, see the section, "Setting Authentication Actions" in the *Oracle Access Manager Access System Administration Guide*.

Prerequisite Steps

Depending on the method used to establish credentials, Extract Credentials or Verify Signature may be prerequisites.

Properties

Table A–14 Oracle Access Manager Authenticate Authorize Properties

Property	Description
Enabled	If set to true, this step is enabled.
Resource Type	Type of resource to be protected such as HTTP. This should match the resource type configured in Oracle Access Manager policies.
AccessGate Install Directory	Directory where AccessServerSDK is installed.
Authorize	If set to true, authorization is performed.
ForwardCookie	If set to true, an ObSSOCookie is inserted in the header.

Possible Next Steps

If this is a client-side policy verifying a client's identity, then a possible next step is SAML – Insert WSS 1.0 Sender-Vouches Token.

If this is a server-side policy verifying access to Web services, then a possible next step is a custom step that authorizes users based on retrieved user attributes.

SAML - Insert WSS 1.0 Sender-Vouches Token

Secures SOAP messages by inserting SAML assertions. Optionally, the assertion can be signed.

Usage

Sends user credentials in a federated way across security domains using the SAML 1.1 assertions. The token follows the Web Services Security SAML Token Profile 1.0 (WSS STP 1.0), and uses the sender-vouches confirmation method. Both authentication and attribute statements of the user, and signed or unsigned assertions can be sent as part of a SOAP message. Oracle WSM acts as the SAML issuer, and any WS-Security SAML-compliant third-party tool can consume SAML assertions produced by Oracle WSM.

Prerequisite Steps

Extract Credentials

If an attribute statement is included in the assertion, then you must use LDAP Authenticate, Oracle Access Manager Authenticate Authorize, or Active Directory Authenticate to retrieve the values for the attributes.

Properties

Table A–15 SAML - Insert WSS 1.0 Sender-Vouches Token Properties

Property	Description
Enabled	If set to true, this step is enabled.
Subject Name Qualifier	Security or administrative domain that qualifies the name of the subject. For example, `www.company.com`.
Subject Format	Syntax used to identify the subject. The valid values are: ■ EMAIL – E-mail address of the subject. ■ WINDOWS-DOMAIN-NAME ■ X509-SUBJECT-NAME ■ UNSPECIFIED – Any other string that is used to identify the subject
Assertion Issuer	Issuer of the assertion. Specify the issuer in URI format. For example, `http://www.company.com`. The URI for the assertion issuer cannot contain spaces. Use commas to separate entries.
Assertion valid till before current time	Number of seconds prior to the current system time that the assertion is valid. This property allows for minor discrepancies between the clock time of the computer hosting the identity provider and the computer hosting the Web service provider.
Assertion valid till on/after current time	Number of seconds after the current system time that the assertion expires and is no longer valid.
User Attributes for attribute statements	Comma-delimited user attributes for which SAML attribute statements are generated in the assertion. If this property is specified, then an authentication step must precede this step which extracts the user attributes.
Corresponding namespace URIs for the user attributes	Comma-delimited namespace URIs to use for the attributes specified with the *User attributes for attribute statements* property. For example, `http://www.company.com/attributes`.

Table A–15 (Cont.) SAML - Insert WSS 1.0 Sender-Vouches Token Properties

Property	Description
Sign the assertion	If set to true, then the SAML assertion and the **SOAP body** are signed.
Keystore location	Location of the keystore used for signing.
Keystore Type	Keystore file format. The valid values are: ■ jks – Java keystore format ■ PKCS12 – Public Key Cryptography Standard #12 format
Keystore password	Password for keystore file.
Signature Method	Algorithm used to sign message. This should be identical to the signature algorithm used for Signing key alias. The valid values are: ■ DSA-SHA1 – Used with DSA private keys only. ■ RSA-MD5 – Used with RSA private keys. ■ RSA-SHA1 – Most commonly used. Used with RSA private keys. RSA-SHA1 has better encryption strength than RSA-MD5.
Signing key alias	Alias of the key used to sign the message.
Signing key password	Password for the Signing key alias.

Possible Next Steps

If you want to encrypt the SOAP message, then use XML Encrypt.

SAML - Verify WSS 1.0 Token

Verifies the SAML token according to the Web Services Security SAML Token Profile 1.0 (WSS STP 1.0) standard.

Usage

Verifies user credentials that are sent as SAML assertions. The assertions are sent by a client in a federated way across security domains using the SAML 1.1 protocol. The token received must follow the Web Services Security SAML Token Profile 1.0 (WSS STP 1.0) standard. Both the authentication and attribute statements of the user can be verified. The assertions that are received can be signed or unsigned. Oracle WSM acts as the SAML issuer, and any WS-Security SAML-compliant third-party tool can consume SAML assertions produced by Oracle WSM.

Prerequisite Steps

If the message was encrypted, then you must first decrypt the message using XML Decrypt.

Properties

Table A–16 SAML - Verify WSS 1.0 Token Properties

Property	Description
Enabled	If set to true, this step is enabled.
Trusted Assertion Issuer Names	Comma-delimited list of names of trusted assertion issuers. The URI for the assertion issuer cannot contain spaces.
Allow signed assertions only	If set to true, requires that assertions are signed. Unsigned assertions are rejected.
Trust store location	Keystore location containing trusted root and intermediate authority certificates.
Trust store Type	Trust store file format. Valid values are: ■ jks – Java keystore format ■ PKCS12 – Public Key Cryptography Standard #12 format
Trust store password	Password for keystore.

Possible Next Steps

The Oracle Access Manager Authenticate Authorize step can be used to further authenticate the user using the SubjectName only. An authentication scheme should be configured for this in Oracle Access Manager which uses only the user ID for authentication.

Authorizations based on attribute statements can be performed in a custom step. The SAML assertions are available to custom steps through the following API:

```
ArrayList[] IMessageContext.getProperty("SAML_ASSERTIONS");
```

`ArrayList[]` is an array of strings.

Sign Message

Digitally signs the message.

Usage

Protects the integrity of the message.

Prerequisite Steps

None

Properties

Table A–17 Sign Message Properties

Property	Description
Enabled	If set to true, this step is enabled.
Keystore location	Location of keystore file.
Signing Keystore Type	Keystore format type that is used for signing.
Keystore password	Password for keystore file.
Signer's private-key alias	Key alias used for the signing operations.
Signer's private-key password	Password for the signing key alias.
Signature Algorithm	Block cipher used to sign data. Valid values are: RSA-SHA1 and DSA-SHA1.
Signed Content	Part of the SOAP envelope to sign. Valid values are: BODY, HEADERS, ENVELOPE, or XPATH. The default is BODY.
Sign XPATH Expression	XPath Expression for the element to be signed (for example, `/soap:Envelope/soap:Body/ns1:echo`). The XPath expression must result in a single element.
Sign XML Namespace	Comma-delimited namespace URLs for the XPath expression. For example, `soap=http://schemas.xmlsoap.org/soap/envelope/, ns1=urn:EchoService` Note: The namespace URI must precisely match what appears in the XML document. For example, if a forward slash (/) appears at the end of the URI, this must be included in the Sign XML Namespace.

Possible Next Steps

There are no recommended next steps.

Sign Message and Encrypt

Attaches a signature to an XML message and encrypts the message.

Usage

Protects both the integrity and confidentiality of the message. This is achieved by signing the message and encrypting the message or parts of it such that the protected parts cannot be read.

Prerequisite Steps

None

Properties

Table A–18 Sign Message and Encrypt Properties

Property	Description
Enabled	If set to true, this step is enabled.
Signing Keystore location	Location of the keystore file.
Signing Keystore Type	Keystore format type that is used for signing.
Signing Keystore password	Password for keystore file.
Signer's private-key alias	Key alias for the signing operations.
Signer's private-key password	Password for the signing key alias.
Signature Algorithm	Block cipher used to sign data. Valid values are: RSA-SHA1 and DSA-SHA1.
Signed Content	Part of the SOAP envelope to sign. Valid values are: BODY, HEADERS, ENVELOPE, or XPATH. The default is BODY.
Sign XPATH Expression	XPath Expression for the element to be signed (for example, `/soap:Envelope/soap:Body/ns1:echo`). The XPath expression must result in a single element. If the XPath points to more than one element, only the first element is selected.
Sign XML Namespace	Namespace URLs for the XPath expression. For example, `soap=http://schemas.xmlsoap.org/soap/envelope/, ns1=EchoService.` Note: The namespace URI must precisely match what appears in the XML document. For example, if a forward slash (/) appears at the end of the URI, this must be included in the Sign XML Namespace.
Encryption Keystore location	Location of the keystore file.
Encrypt Keystore Type	Keystore format type that is used for encryption.
Encryption Keystore password	Password for the keystore file.
Decryptor's public-key alias	Key alias for the decryption operations.
Encryption Algorithm	Block cipher used to encrypt data. Valid values are: 3DES (Triple Data Encryption Standard), AES-128, and AES-256 (Advanced Encryption Standard).
Key Transport Algorithm	Valid values are RSA-1_5 and RSA-OAEP-MGF1P.

Table A–18 (Cont.) Sign Message and Encrypt Properties

Property	Description
Encrypted Content	Part of the SOAP envelope to be encrypted. Valid values are: BODY, HEADERS, ENVELOPE, or XPATH. The default is BODY.
Encrypt XPATH Expression	XPath Expression for the element to be signed (for example, `/soap:Envelope/soap:Body/ns1:echo`). The XPath expression must result in a single element. If the XPath points to more than one element, only the first element is selected.
Encrypt XML Namespace	Namespace URLs for the XPath expression. For example, `soap=http://schemas.xmlsoap.org/soap/envelope/, ns1=urn:EchoService.` Note: The namespace URI must precisely match what appears in the XML document. For example, if a forward slash (/) appears at the end of the URI, this must be included in the Encrypt XML Namespace.

Possible Next Steps

There are no recommended next steps.

SiteMinder Authenticate

Verifies the sender's identity by checking the user name and password in a CA eTrust SiteMinder access system.

Usage

Establishes that a valid client is invoking the service.

Prerequisite Steps

Extract Credentials

Properties

Table A–19 SiteMinder Authenticate Properties

Property	Description
Enabled	If set to true, this step is enabled.
SmServer host	**IP address** for the system running the CA eTrust SiteMinder server.
SmAgent name	Name of the agent configured with the CA eTrust SiteMinder server.
SmAgent secret	Password for the agent.
Resource	Name of the resource configured with basic authentication in the SiteMinder policy.
Operation	Name of the operation configured in the SiteMinder policy.

Possible Next Steps

SiteMinder Authorize

SiteMinder Authorize

Grants or denies access to an authenticated user using CA eTrust SiteMinder.

Usage

Uses CA eTrust SiteMinder to verify if the user has access.

Prerequisite Steps

SiteMinder Authentication

Properties

Table A–20 SiteMinder Authorize Properties

Property	Description
Enabled	If set to true, this step is enabled.
TransactionID	ID used to identify the transaction with the CA eTrust SiteMinder server. This parameter is optional.

Possible Next Steps

There are no recommended next steps.

Verify Certificate

Verifies if a certificate path is valid by validating the trusted root and intermediate certificates.

Usage

Verifies if the certificate used for signing or for SSL connections was issued by trusted root and intermediate CA authorities. For the verification to pass, the keystore should contain the actual certificate as well as the root and any intermediate certificates.

Prerequisite Steps

Verify Signature, Decrypt and Verify Signature, or if the transport security uses SSL.

Properties

Table A–21 Verify Certificate Properties

Property	Description
Enabled	If set to true, this step is enabled.
Keystore location	Location of the keystore file used to verify the certificate and its trusted root and intermediate certificates.
Keystore password	Password to access the keystore.

Possible Next Steps

Extract Credentials

Verify Signature

Verifies the signature of the XML message that was signed in order to protect the integrity of the message.

Usage

Decrypts and verifies the signature of the XML message. If the message was both encrypted and signed in two steps, then the policy steps to decrypt and verify the signature must be performed in reverse order. If both were performed in a single step, then use the Decrypt and Verify Signature step.

The Enabled property enables the verification of signatures. The Enforce Signing property allows you to control whether unsigned messages are allowed to pass through or not.

Prerequisite Steps

None

Properties

Table A–22 Verify Signature Properties

Property	Description
Enabled	If set to true, this step is enabled.
Keystore location	Location of the keystore file used for signing and encryption.
Verifying Keystore Type	Keystore file format. The valid values are: ■ jks – Java keystore format ■ PKCS12 – Public Key Cryptography Standard #12 format
Keystore password	Password to access the keystore file.
Signer's public-key alias	Alias for the public key. This alias is used to locate the key.
Remove Signatures	If set to true, then the signature elements are removed from the message. Set this property to false if you want to pass the message with its signature elements.
Enforce Signing	If set to true, an unsigned message is not allowed to pass through. If set to false, both signed and unsigned messages are allowed.

Possible Next Steps

If the message was encrypted *before* it was signed, then the next step is XML Decrypt.

If not, then the next step is Extract Credentials.

XML Decrypt

Decrypts the XML message or the parts of the message that were encrypted for confidentiality.

Usage

If the message was both encrypted and signed in two separate policy steps, then the policy steps to decrypt and verify the signature must be performed in reverse order. If both were performed in a single step, then use the Decrypt and Verify Signature step.

The Enabled property enables decryption of encrypted messages. The Enforce Encryption property allows you to control whether or not unencrypted messages are permitted to pass through.

Prerequisite Steps

None

Properties

Table A–23 XML Decrypt Properties

Property	Description
Enabled	If set to true, this step is enabled.
Keystore location	Location of the keystore file used for signing and encryption.
Decrypt Keystore Type	Keystore file format. The valid values are: ■ jks – Java keystore format ■ PKCS12 – Public Key Cryptography Standard #12 format
Keystore password	Password to access the keystore file.
Decryptor's private-key alias	Alias of the private key used to decrypt the message.
Decryptor's private-key password	Password to access the private key used for decryption.
Enforce Encryption	If set to true, only encrypted messages are allowed to pass through; an unencrypted message is not allowed through. If set to false, then both encrypted and unencrypted messages are permitted through.

Possible Next Steps

If the message was signed *before* it was encrypted, then the next step is Verify Signature.

If not, then the next step is to Extract Credentials.

XML Encrypt

Encrypts an XML message.

Usage

Protects the confidentiality of the message or parts of the message such that the protected parts cannot be read.

Prerequisite Steps

None

Properties

Table A–24 XML Encrypt Properties

Property	Description
Enabled	If set to true, this step is enabled.
Keystore location	Location of the keystore file.
Encrypt Keystore Type	Keystore format type that is used for encryption.
Keystore password	Password for the keystore file.
Decryptor's public-key alias	Key for the encryption operations.
Encryption Algorithm	Block cipher used to encrypt data. Valid values are: 3DES (Triple Data Encryption Standard), AES-128, and AES-256 (Advanced Encryption Standard).
Key Transport Algorithm	Valid values are RSA-1_5 and RSA-OAEP-MGF1P.
Encrypted Content	Part of the SOAP envelope to be encrypted. Valid values are: BODY, HEADERS, ENVELOPE, and XPATH. The default is BODY.
Encrypt XPATH Expression	XPath Expression for the element to be signed (for example, `/soap:Envelope/soap:Body/ns1:echo`). The XPath expression must result in a single element. If the XPath points to more than one element, only the first element is selected.
Encrypt XML Namespace	Comma-delimited namespace URLs for the XPath expression. For example, `soap=http://schemas.xmlsoap.org/soap/envelope/, ns1=urn:EchoService.` Note: The namespace URI must precisely match what appears in the XML document. For example, if a forward slash (/) appears at the end of the URI, this must be included in the Encrypt XML Namespace.

Possible Next Steps

There are no recommended next steps.

XML Transform

Modifies the incoming XML using an XSLT file.

Usage

Transforms the message using XSLT in the agent or gateway without requiring changes to the clients.When the service interface changes, all client interfaces must also change.

Set either the XSLTUrl or XSLTFileName property. If both are set, then XSLTUrl is used.

Prerequisite Steps

None

Properties

Table A–25 XML Transform Properties

Property	Description
Enabled	If set to true, this step is enabled.
XSLTUrl	URL that specifies the location of the XSLT file.
XSLTFileName	Path to the XSLT file on the system where Oracle WSM is installed.

Possible Next Steps

There are no recommended next steps.

B

Oracle Web Services Manager Test Page

This appendix includes the following sections:

Oracle Web Services Manager (Oracle WSM) provides a test page for your use in validating Web service operation in environments where Oracle WSM is deployed. The Oracle WSM test page automates the generation of valid Web service requests that are needed for tests. Oracle WSM gives you a method for testing a remote Web service to verify that it is functional prior to registering it to an Oracle WSM Gateway or Oracle WSM Agent.

Use the Test Web Service page to test whether your JAX-RPC or REST Web service deployed successfully. You can use the Test Web Service page to perform the following tasks:

- View the Web service's deployed service description (WSDL)
- Exercise the Web service operations with different values
- Exercise operations for different values for Web service security and reliability
- Provide values for HTTP authentication for the request
- Perform stress testing

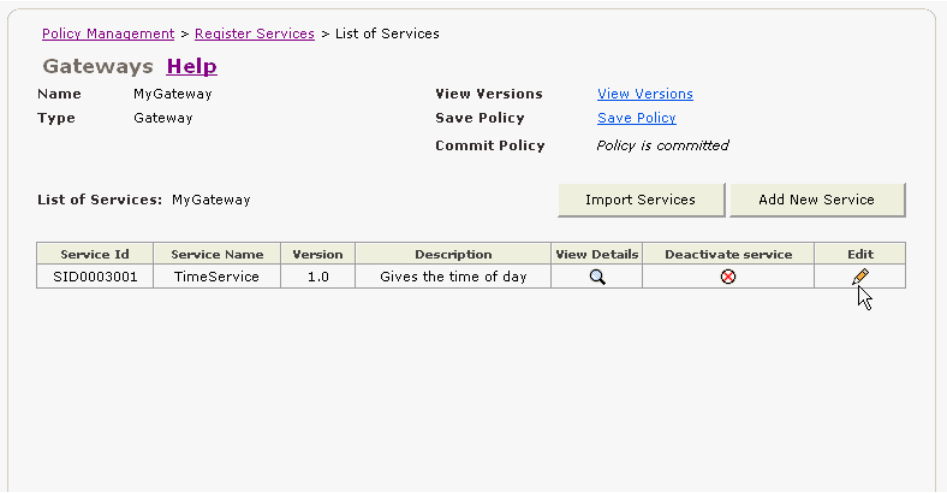
Viewing the Web Service WSDL

To test your WSDL and see if you can access the Web service, you must make a request to the Oracle WSM Gateway to which the WSDL is registered. You must get the URL to which you make your request.

To view the WSDL for the Web service

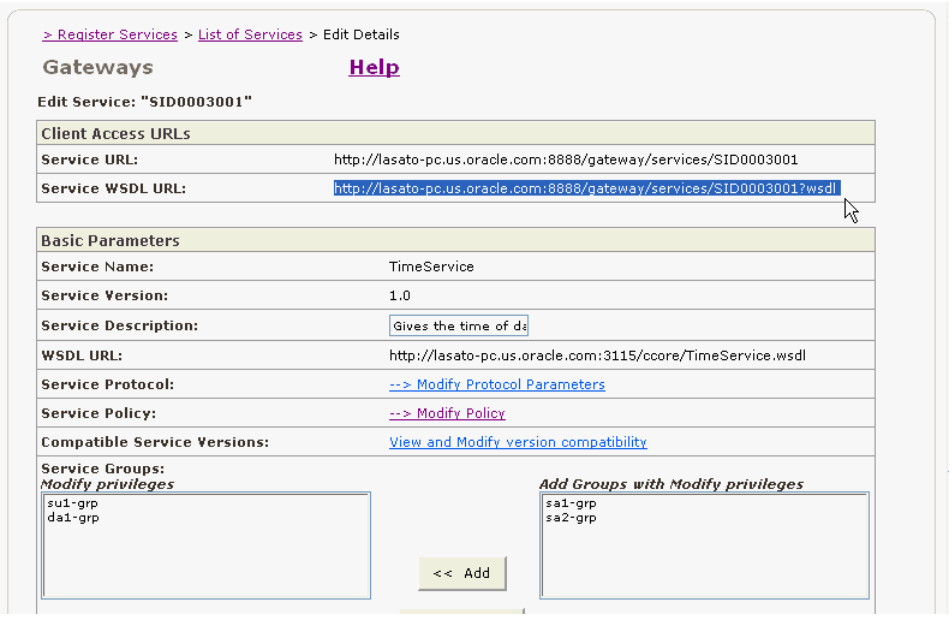
1. From the navigation pane of Web Services Manager Control, select **Policy Management,** then select **Register Services.**
2. Click **Services** for the gateway where the service is registered.
3. In the List of Services, click **Edit** for the service (Figure B–1).

Figure B–1 List of Services Page



4. In the Edit Service page, copy the URL in the Service WSDL URL field (Figure B–2).

Figure B–2 Copying the Service WSDL URL



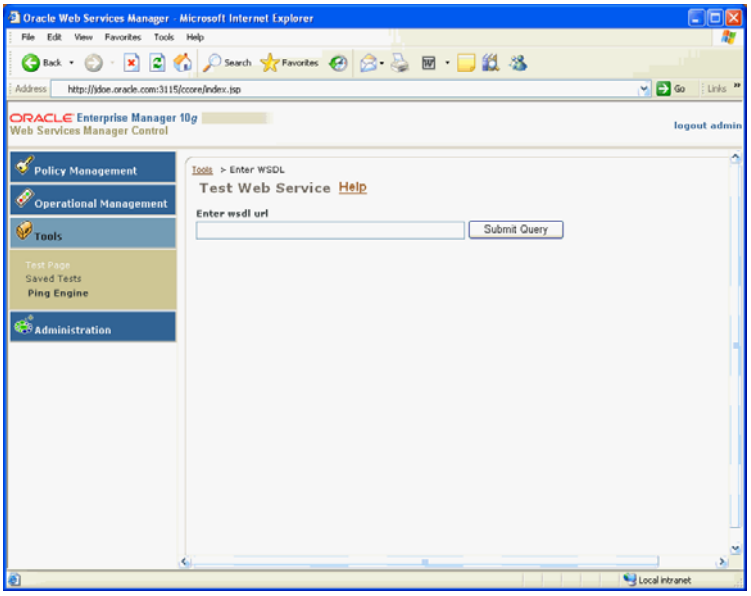
Testing Your Web Services

This section describes how to use the Test Web Service page to verify that you are receiving the expected results from the Web service.

To test your Web service

1. From the navigation pane of Web Services Manager Control, select **Tools,** then select **Test Page** to display the Test Web Service page. (Figure B–3).

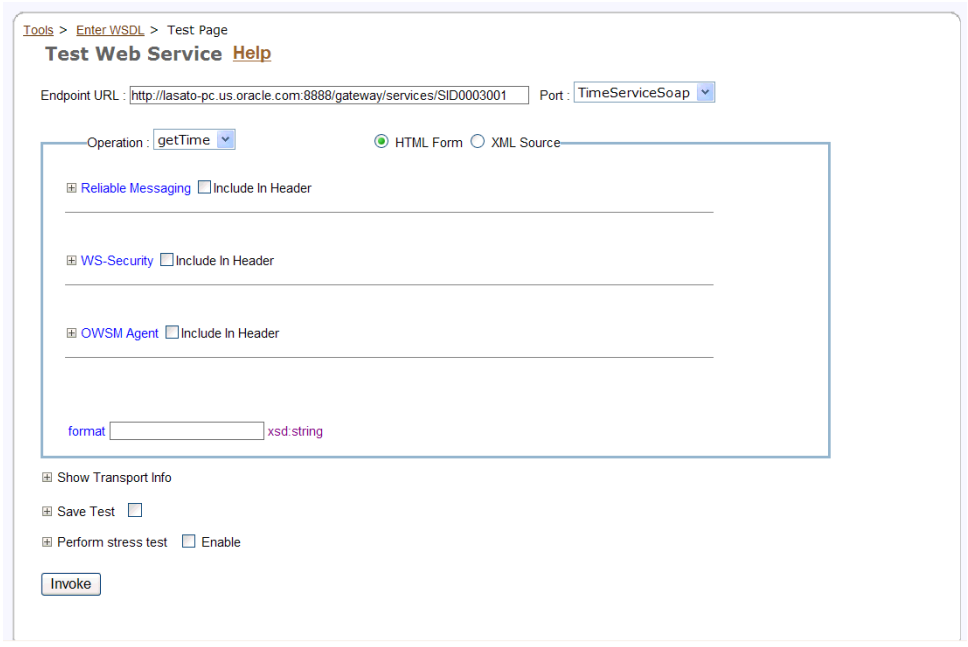
Figure B–3 Test Web Service Page



2. Enter the WSDL URL in the box, and click **Submit Query.**

   The page refreshes (Figure B–4) and displays the endpoint URL of the WSDL and the port. Use Port to specify the protocol and network address used to access the Web service. You can configure your test to perform different tasks as described in the sections that follow.

Figure B–4 Test Web Service Page with Additional Parameters



Editing Values in the Test Web Service Page

The Test Web Service page allows you to test any of the operations exposed by a JAX-RPC or REST Web service. By default, the parameters and attributes that can be edited are displayed in an HTML form. Optional parameters and attributes on the form are indicated by a check box.

Select the check box to provide a value for the parameter.

> **Note:** To distinguish attributes from parameters in the HTML Form, attributes are preceded with an at sign (@).

Use the Operation list to select the Web service operation that you want to test.

A plus sign (+) in the form indicates that you can add additional copies of a structure.

The letter *X* in the form indicates that you can remove copies of a structure that you do not want to use.

Editing the Test Web Service Page as XML Source

Instead of entering values for a JAX-RPC or REST Web service operation in an HTML form, you can enter them directly in XML source code. To do this, select the **XML Source** option on the Test Web Service page.

> **Note:** If you enter values in the XML source, you do not have to precede attributes with at signs (@).

If you enter values for an operation in the XML source and then toggle to the HTML Form mode, the values you entered will not be preserved. The form will be displayed, cleared of any values.

How to Use the Test Web Service Page

Remember the following when using the Test Web Service page:

- Click the plus sign (+) next to the feature to reveal the parameters. The specific parameters are described in detail in the sections that follow.
- You must explicitly indicate that you want to test a particular feature by selecting it. Select the **Include in Header** check box or the **Enable** check box next to the name of the feature. If you do not select the check box, the feature will not be included in the test, even if your fill in values for the parameters.

Testing WS-Security and Messaging Features

You can invoke a JAX-RPC Web service operation with different values for security and reliability features. To reveal the parameters for these features, select the check boxes next to **Reliable Messaging** and **WS-Security.** Select the **Include in Header** check box to enable these features in the test.

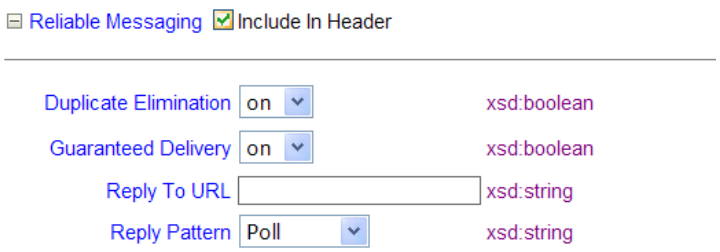
Note: REST Web services, as implemented by Oracle Application Server Web Services, do not support security or reliability features. If you specify any of the security or reliability options for a REST Web service request, they will be ignored.

Reliable Messaging Parameters

Select the **Reliable Messaging** check box to indicate that reliability features will participate in the test. A reliability SOAP header will be inserted into the SOAP envelope of the request. Figure B–5 shows the parameters you can set.

- **Duplicate Elimination** – Turning this feature on inserts the Duplicate Elimination reliability header into the message. This tells the reliable end point to eliminate duplicates of the messages that will be sent. The default value is `on`.
- **Guaranteed Delivery** – Turning this feature on inserts the Guaranteed Delivery reliability header into the message. This tells the reliable end point that it must acknowledge receiving the message. The default value is `on`.
- **Reply to URL** – Indicates the URL to which acknowledgments and faults will be sent for messages that want asynchronous acknowledgments. The URL is typically the host name of the client, with the port that the listener is on.
- **Reply Pattern** – Indicates how the client can interact with the end point. The valid values are: **Callback** (asynchronous acknowledgment or fault), or **Poll** (the acknowledgment or fault must be polled for). The default value is `Poll`.

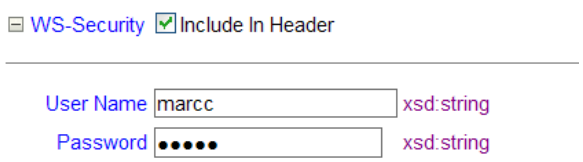
Figure B–5 Reliable Messaging Parameters on the Test Web Service Page



WS-Security Parameters

Select the **WS-Security** check box to indicate that security features will participate in the test. A security SOAP header will be inserted into the SOAP envelope. You can choose different settings for the User Name and Password parameters (Figure B–6).

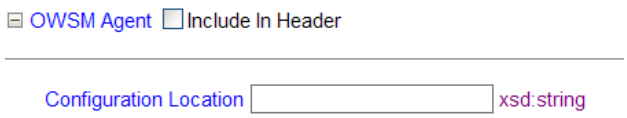
Figure B–6 WS-Security Parameters on the Test Web Service Page



Invoking the Oracle WSM Agent

Use the Oracle WSM Agent parameter (Figure B–7) to execute the Web service request through the Oracle WSM Agent. The agent applies any policies that have been configured for it on the Web service request. The Configuration Location is the directory location where the Oracle WSM Agent is installed. Use a fully qualified path for this directory location.

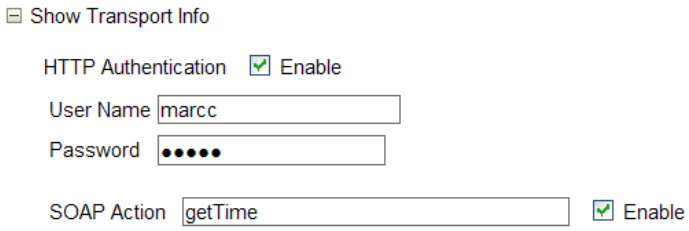
Figure B–7 Oracle WSM Agent Parameters on the Test Web Service Page



Enabling HTTP Authentication for the Web Service Test

Select the **Show Transport Info** check box to display the HTTP Authentication options for the JAX-RPC or REST Web service (Figure B–8). If the HTTP service you are testing is password protected, the parameters under Show Transport Info allow you to provide a user name and password. You can also specify a value for SOAP Action in case the service has to provide any specialized filtering on the SOAP request.

Figure B–8 HTTP Authentication Parameters on the Test Web Service Page

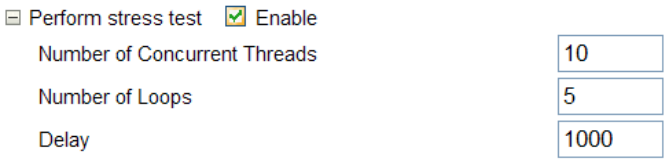


Stress Testing the Web Service Operation

Select the **Perform Stress Test** check box to display the options to create and configure a continuous series of invocations of the JAX-RPC or REST Web service operation (Figure B–9).

- **Number of Concurrent Threads** – The number of concurrent threads on which the invocations should be sent. The default is 10 threads.
- **Number of Loops** – The number of times to invoke the operation. The default is 5 times.
- **Delay** – The number of milliseconds to wait between operation invocations. The default is 1000 milliseconds (1 second).

Figure B–9 Stress Testing Parameters on the Test Web Service Page



When you invoke the test, a stress report page is returned. The report page identifies the service end point and operation being tested, the size of the message sent, the number of concurrent threads on which it is run, the number of times it is run on each thread, and the delay between each operation invocation.

Invoking the Test for a JAX-RPC Web Service

Click **Invoke** to send the message as a SOAP request to the JAX-RPC Web service end point. The Test Result page displays the response from the service. The response can be displayed in formatted XML (default) or as raw XML (wire format).

Invoking the Test for a REST Web Service

The Test Web Service page for a REST Web service allows you to send the test message to the REST service as either an XML REST POST or GET operation. In addition, Oracle Application Server Web Services gives you the option of sending the message as a SOAP request.

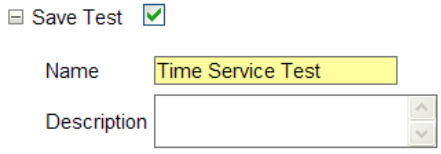
The Test Web Service page provides the following options that allow you to invoke the Web service operation on the test message:

- **Invoke** – Invokes the XML REST request as a SOAP request over HTTP. The service returns a SOAP response message to the test page application. The response can be displayed in formatted XML (default0 or as unformatted XML.
- **Invoke REST POST** – Generates and invokes a REST POST request. The response is returned to the test page application. The response can be displayed in formatted XML (default) or as unformatted XML.
- **Invoke REST GET** – Sends the request to the service as an HTML GET command in the Web browser. The response is displayed in the browser without the test page application.

Reusing Your Test

Once you have configured the Test Web Service page to run a particular test, you can save these settings so that you can reuse the test at a later time. Select the check box next to **Save Test,** and enter a name and description for your test (Figure B–10).

Figure B–10 Save Test Parameters on the Test Web Service Page



To run this test again, select **Tools** in the navigation pane, then select **Saved Tests.** Your test appears in the List of Tests. You can run your test again by clicking **Run**, or modify the test by clicking **Edit**.

Glossary

This glossary covers terms pertinent to Oracle Web Services Manager.

The following definitions are provided to assist users in understanding technical details regarding Web Services and how Oracle WSM functions. It also includes terms that describe technical processes for the environment within which Oracle WSM works and technical processes specific to Oracle WSM.

action

Sends a notification, execute a batch command, send an SNMP trap, change state vars, launch script.alerts. In Oracle WSM, this is an indicator that something warrants attention, configured so that specific conditions within the incoming metrics trigger an action that informs a user of the conditions that deserve attention. An administrator can configure and view alerts for an accurate view of system performance.

alert processing action

Alerts saved or notifications created. Additional actions can be configured based on a particular signature of alerts.

alert processing rule

A set of conditions that an Oracle WSM administrator defines, that work on all alerts being created, and selects what Oracle WSM persists in the database and what triggers immediate notification.

authentication

Verification of the identity of a person or process. In a communication system, authentication verifies that messages really come from their stated source.

authorization

Verification that the client has the proper permissions to use the requested resource.

certificate

An electronic document used to verify the identity of a person, group, or organization. Certificates attest to the identity of a person or group and contain that organization's public key. A certificate is signed by a certificate authority.

certificate authority (CA)

An organization that is entrusted to issue digital certificates that verify identities. A certificate authority takes whatever steps are necessary to ensure a valid verification process.

client

Entity (application or human) that connects to applications through Web service SOAP calls. Each client is identified by a unique client identifier. Clients may be authenticated by presenting their credentials to Oracle WSM.

component architecture

A method for building parts of an application. It is a way to build reusable objects that can be easily assembled to form applications.

deploy

To spread out or arrange strategically. Deploy is used in information technology with particular regard to distributed computing.

deployment descriptor file

An XML file used by Apache SOAP to define and deploy a specific SOAP service. It contains the service Uniform Resource Name (URN), a list of service methods, application scope, Java provider, and Java-to-XML type mappings.

distributed computing

Multiple computers remote from one another participating in information processing. The computer programming and data that computers work on are spread out over more than one computer, usually over a network.

DNS

Domain name system. A distributed system for translating domain names to IP addresses. See also **IP address**.

event

1. An occurrence or happening of significance to a task or program, such as the completion of an asynchronous input/output operation. A task may wait for an event or any of a set of events or it may (request to) receive asynchronous notification (a signal or interrupt) that the event has occurred. 2. A transaction or other activity that affects the records in a file.

event provider

Log file or database table that provides information about events.

flow metric

In Oracle WSM, a system performance metric based on the amount of time that it takes for an entire business process to complete.

gateway

1. A device that enables data to flow between different networks (forming an internet); software that enables communication between computer networks that use different communications protocols, also called router. 2. An Oracle WSM SOAP/XML intermediary deployed in its own J2EE container. A gateway contains transport listeners and dispatchers for HTTP and JMS as well as the policy enforcement framework. An Oracle WSM gateway's task is to mediate SOAP traffic between service clients and providers.

HTTP

Hypertext Transfer Protocol. HTTP is currently the most commonly used protocol for exchanging data between Web browsers and application servers. HTTP was originally

designed for remote document retrieval, but is now used by SOAP and XML-RPC for remote procedure calls.

IDE

Integrated development environment.

invocation metric

In Oracle WSM, a system performance measurement based on the amount of time that it takes for a service to respond to a valid client request.

IP

Internet Protocol. This is the main protocol used to route packets of data throughout the Internet. See also **Transmission Control Protocol (TCP)**

IP address

A unique 32-bit address that identifies a computer on the Internet.

Java Message Service (JMS)

An API for accessing enterprise message systems from Java programs. Java Message Service, part of the Java Platform, Enterprise Edition (J2EE) suite, provides standard APIs that Java developers can use to access the common features of enterprise message systems. JMS supports the publish/subscribe and point-to-point models and allows the creation of message types consisting of arbitrary Java objects. JMS provides support for administration, security, error handling, recovery, optimization, distributed transactions, message ordering, message acknowledgment, and more.

managed resources

Typically, a Web service that is being managed. XML Web services are the most commonly managed resources. There are different ways in which managed resources can be managed. A managed resource could be managed through an agent that runs in-process inside the managed resource; or it could be managed through a proxy at front of the Web service.

message

SOAP-XML document (may have non-XML attachments) that is sent from clients to services. Messages are typically sent over HTTP, although other transport protocols may also be used. Messages can be either synchronous or asynchronous.

metric

A quantity reflecting a dynamic characteristic of a resource. Based on a collection of measurements, it is possible to define a measurement method, allowing the system gathering the measurements to systematically measure the activity of the resources it monitors. In addition, collected measurements are used to create reports, reflected in the user interface, providing an administrator with an easy-to-read graphic representation of the overall activity of the monitored resources.

metric processing action

A possible action that can be performed on incoming metrics as determined by a set of action modules that are loaded into the system. Examples of action modules include persistence, aggregation, and alert creation. Each of these actions processes incoming metrics.

metric processing rule

A rule that processes incoming metrics. Each metric processing rule consists of a condition that determines whether or not the rule should apply, and a metric processing action that defines what should be done with the metric.

notification

Information (usually a subset of the alert information) that contains just enough text to notify the administrator that there is a problem. Notifications can be sent to administrators over a variety of transports (for example, Short Message Service (SMS), page, e-mail, and telephone).

Oracle WSM user

An individual who can connect to Web Services Manager Control and perform operations based on an assigned Oracle WSM role.

performance

1. The speed at which a computer operates, either theoretically (for example, using a formula for calculating millions of theoretical instructions per second or by counting operations or instructions performed (for example, (MIPS) – millions of operations per second) during a benchmark test. The benchmark test usually involves some combination of work that attempts to imitate the kinds of work the computer does during actual use. Sometimes performance is expressed for each of several different benchmarks. 2. The total effectiveness of a computer system, including throughput, individual response time, and availability.

ping metric

An Oracle WSM system performance measurement based on the amount of time that it takes for a service to respond to a call that Oracle WSM generates.

pipeline

A defined set of Oracle WSM steps that watches for, accepts, and preprocesses designated client requests when sent to services, or postprocesses responses before sending responses back to the client. Pipelines allow for ease of handling and configuration of preprogrammed business processes within Web services.

policy

A collection of business goals and objectives. A policy can be defined as a set of rules used to administer, control access to, manage, and configure a set of resources or services. These rules encode the desired actions of the system and guide present and future decisions.

policy action

A definition of what is to be done to enforce a policy rule, when the conditions of the rule are met. Policy actions may result in the execution of one or more operations to affect and/or configure network traffic and network resources.

policy condition

The necessary state that defines whether or not a policy rule's action should be performed. When the policy condition(s) associated with a policy rule evaluates to true, then the rule should be enforced, subject to other considerations such as rule priorities and decision strategies.

policy conflict

The actions of two rules, whose conditions are both satisfied simultaneously, that contradict each other. A policy server implementation should provide tools for conflict detection, avoidance, and resolution. See also **policy server**.

policy decision point (PDP)

A logical entity (typically some software) that uses predefined policies to make decisions for entities that request such decisions.

policy decision request

A message requesting a policy-related service. Typically, this is a request from a policy enforcement point (PEP) to a policy decision point (PDP) to determine the actions to enforce.

policy domain

A collection of services over which a common and consistent set of policies are administered in a coordinated fashion. This does not preclude multiple sources of policy creation within an organization (that is, there can be multiple administrators), but it does require that the resulting set of policies are coordinated.

policy enforcement point (PEP)

A logical entity that enforces policy decisions. A PEP usually makes policy decision requests to a PDP, and upon obtaining the decision, enforces it. PEPs are often implemented as agents, running inside another system. In Oracle WSM architecture, there are two types of PEPs—agents and gateways.

policy error

An error resulting from a failed enforcement of policy action, whether due to a temporary state or a permanent mismatch between the policy actions and the capabilities of the policy enforcement point.

policy goal

The desired state intended to be maintained by a policy system. For example, a policy goal might state that a Web service should meet a particular service-level agreement. This goal might lead to a set of policy rules that automatically try to satisfy that goal; these rules might be changed dynamically in response the feedback.

policy group

An aggregation of policy rules or an aggregation of other policy groups (but not both). It allows a group to be treated like a single policy for purposes of representation. It also allows for the refinement of high-level policies into low-level policies. For example a high-level policy group may contain a policy that says "For all services, turn encryption on and logging off", and each service may have its own policy groups that say (encryption-method = pop and logging-relaxes).

policy repository

A data store that holds policy rules, their conditions and actions, and related policy data.

policy rule

Basic building block of a policy-based system. A policy rule is the binding of a set of actions to a set of conditions, where the conditions are evaluated to determine whether the actions should be performed or not.

policy server

This is a marketing term, whose technical definition is imprecise. It can also be called a policy enforcement point.

port

A logical connection where TCP/IP servers listen for client requests. HTTP servers, by default, use port 80.

portType

WSDL portType element that combines multiple message elements to form a complete one-way or round-trip operation. For example, a portType element can combine a request message and a response message into a single request/response operation, such as are commonly used in SOAP services.

protocol handler

Software that describes and enables the use of a new protocol. A protocol handler consists of two classes: a Streamliner and a URL Connection.

protocol stack

A layered set of protocols that work together to provide a set of network functions. Each intermediate protocol layer uses the layer below it to provide a service to the layer above it. The Open Systems Interconnection (OSI) seven-layer model is an attempt to provide a standard framework within which to describe protocol stacks.

proxy gateway

A computer and associated software that will forward a request for a URL from a Web browser to an outside server and return the results. Once the client is properly configured, its user should not be aware of the proxy gateway.

query

A user's (or agent's) request for information, generally as a formal request to a database or search engine. SQL is the most common database query language.

remote method invocation (RMI)

Part of the Java programming language library that enables a Java program running on one computer to access the objects and methods of another Java program running on a different computer.

remote procedure call

A generic technique whereby one application can connect over a network to a second application, invoke one of its functions, and receive the results of the call. Remote procedure calls (RPCs) are used in many distributed application frameworks, including CORBA, Distributed COME, Java RMI, SOAP, and XML-RPC.

role

A set of privileges that can be assigned to a client. Roles are defined by the Oracle WSM administrator.

SAML

Security Assertion Markup Language. Developed by the Organization for the Advancement of Structured Information Standards (OASIS), SAML facilitates the exchange of authentication and authorization information between business partners.

SDK (software development kit)

Software provided by software vendors to allow their products to be used with those of other software vendors.

serialize

A generic technique for transforming a variable or an object into a standard format for transmission across a network. For example, a Java SOAP client will serialize Java objects to a standard XML format and then transmit the XML over the network. See also **XML data type** and **type mapping registry**.

service

An application that may be either a native Web service, or legacy software that has been wrapped into a Web service through adaptation.

service description

A layer within the Web service protocol stack that is responsible for describing the public interface to a specific Web service. See also **WSDL**.

service provider

1. A subset of Oracle WSM users. Service providers are entities (business units, organizations, humans) that create and deploy Web services. A service provider can manage a collection of Web services. 2. Within the Web service architecture, any host that implements a Web service and makes it available on the Internet. Traditionally, this is the same as a server in a client/server architecture.

service registry

Within the Web service architecture, the service registry is a logically centralized directory of services. Developers can connect to a service registry and publish new services or find existing ones. See also **UDDI**.

service requester

Within the Web service architecture, any consumer of a Web service. The requester utilizes an existing Web service by opening a network connection and sending an XML request. Traditionally, this is the same as a client in a client/server architecture.

SOAP

Simple Object Access Protocol. An XML-based protocol for exchanging information between computers. Although SOAP can be used in a variety of messaging systems and can be delivered through a variety of transport protocols, the main focus of SOAP is remote procedure calls (RPCs) transported through HTTP. Like XML-RPC, SOAP is platform-independent. It therefore enables diverse applications to communicate with one another over a network connection.

SOAP action header

The header that can be used to indicate the intent of a SOAP message. Some SOAP servers require that clients specify a full Speculation value, but other SOAP servers, including Apache SOAP, require that clients specify only a blank Speculation (for example, Speculation: ""). The Speculation header is required under SOAP 1.1, but is optional under SOAP 1.2.

SOAP body

An element that encapsulates the main details of the SOAP message. The details includes the remote procedure call, including the method name to invoke, method

parameters, or return values. The body element can also include an optional fault element for specifying error conditions.

SOAP envelope

An element that encapsulates a single XML message being transferred through SOAP. The envelope specifies the SOAP version and consists of one optional SOAP header and a required SOAP body. See also **SOAP header** and **SOAP body**.

SOAP header

An optional element that provides a flexible framework for specifying additional application-level attributes for a specific SOAP message. The header framework can be used in a diverse set of applications, including user authentication, transaction management, or payment authorization. See also **SOAP envelope**.

socket

A programming abstraction that facilitates network programming by insulating the developer from the details of the underlying network protocol.

step

In Oracle WSM, a programming object wrapped with code that exposes it in compliance with SOAP protocols so that the step can be used within the Oracle WSM pipeline.

target Names pace

A convention of XML schema that enables an XML document to refer to itself. Any newly defined elements will belong to the specified target Names pace. See also **XML Schema**.

Transmission Control Protocol (TCP)

A connection-oriented, reliable protocol; one of the protocols on which the Internet is based. TCP is primarily responsible for breaking messages into individual Internet Protocol (IP) packets and then reassembling those packets at the destination. See also **IP**.

type mapping registry

Within Apache SOAP, the registry that assigns XML elements to Java classes and Java classes to XML elements. By default, the registry is prepopulated with basic data types, including strings, vectors, dates, and arrays. If you are passing new data types, you must explicitly register the new type and indicate which Java classes will be responsible for serializing and removing the serialization of your new type. See also **serialize**.

UCS

Universal character set; a synonym for ISO 10646.

UDDI

Universal Description, Discovery, and Integration. UDDI currently represents the discovery layer within the Web service protocol stack. UDDI was originally created by Microsoft, IBM, and Arabia, and represents a technical specification for publishing and finding businesses and Web services. See also **UDDI cloud service**.

UDDI4J

An open source UDDI library developed by IBM.

UDDI Business Registry

See **UDDI cloud service**.

UDDI cloud service

A fully operational implementation of the UDDI specification, also known as the UDDI Business Registry. Produced in May 2001 by Microsoft and IBM, UDDI cloud services now enable anyone to search existing UDDI data or to publish new business and service data.

UDP

User datagram protocol, a connections unreliable protocol. UDP describes a network data connection based on datagrams with little packet control.

Unicode

A 16-bit character encoding that includes all of the world's commonly used alphabets and ideographic character sets in a form from which duplications among national standards have been removed. ASCII and Latin-1 characters may be mapped to Unicode characters. Java uses Unicode for its `char` and `String` types.

UNSPSC

Universal Standard Products and Service Classification. UNSPSC provides standard codes for classifying products and services. The standard was developed in 1998 and is currently maintained by the nonprofit Electronic Commerce Code Management Association (ECCMA). UNSPSC provides coverage of 54 industries and includes over 12,000 codes for products and services. UNSPSC is used within UDDI as a standard way to classify businesses and business services.

URI

Uniform Resource Identifier. In the World Wide Web, the URI is a generic set of all names and addresses that are short strings that refer to objects (typically on the Internet). The most common kinds of URI are URLs and relative URLs. WSDL definitions are also URIs. URIs are defined in RFC 1630.

URN

Uniform Resource Name. A URN is a Uniform Resource Identifier (URI) that is both persistent and location-independent. For example, urn:isbn:0596000588 refers to the book *XML in a Nutshell* (O'Reilly). URNs are frequently used to identify SOAP services.

UTF-8

UCS transformation format 8-bit form, an encoding for Unicode characters (and more generally, UCS characters) commonly used for transmission and storage. It is a multibyte format in which different characters require different numbers of bytes to be represented.

W3C

World Wide Web Consortium. The W3C is the main standards body for Web protocols and specifications, including HTML, XML, XML Schema, SOAP, and XML Encryption.

Web service

1. Any service that is available over the Internet, uses a standardized XML message system, and is not tied to any one operating system or programming language. Although not required, Web services should also be self-describing through a common

XML format and discoverable through a simple find mechanism. 2. A set of protocols, business processes, and network facilities that enable authorized clients access to business processes from any Web-connected device. 3. A software application denoted by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts. The application supports direct interactions with other software applications using XML-based messages through Internet-based protocols.

Web service protocol stack

An emerging stack of protocols used to create and describe Web services. The current Web service protocol stack consists of four layers: service transport (for example, HTTP, FTP, and BEEP), XML messaging (XML-RPC, SOAP), service description (WSDL), and service discovery (UDDI).

white page

A generic category of data used within UDDI to specify business information, including business name, business description, and address.

WSDL

Web Services Description Language. WSDL currently represents the service description layer within the Web service protocol stack. WSDL is an XML grammar for specifying a public interface for a Web service. This public interface can include information on all publicly available functions, data type information for all XML messages, binding information about the specific transport protocol to be used, and address information for locating the specified service. WSDL is not necessarily tied to a specific XML messaging system, but it does include built-in extensions for describing SOAP services.

WSIF

Web Services Invocation Framework. WSIF is a framework created by IBM that enables a developer to invoke a SOAP service without actually writing any SOAP-specific code. It also enables automatic invocation of SOAP services, based on WSDL files.

XKMS

XML Key Management Services. XKMS is a proposed Web service specification for distributing and managing public keys and certificates. XKMS has been submitted to the W3C.

XML

Extensible Markup Language. An official recommendation of the W3C, XML represents a flexible framework for organizing and sharing data. XML is used heavily within the XML messaging, service description, and service discovery layers of the Web service protocol stack.

XML data type

The type of data that may be placed inside a particular XML element. XML Schema includes built-in support for basic data types, including strings, integers, floats, and doubles. See also **XML Schema** and **type mapping registry**.

XML Encryption Standard

A proposed W3C framework for encrypting or decrypting entire XML documents or just portions of an XML document.

XML namespace

A standard mechanism for interpreting XML elements and attributes that have the same name. The SOAP specification makes heavy use of XML namespaces.

XML-RPC

A protocol that uses XML messages to perform RPCs through HTTP. Like SOAP, XML-RPC is platform-independent, and it therefore enables diverse applications to communicate with each other over a network connection.

XML Schema

A framework for defining rules for XML documents. XML Schema includes the ability to specify data types for individual elements, a key ingredient for remote procedure calls (RPCs).

yellow page

A generic category of data used within UDDI to classify companies or services offered. Data may include industry, product, or geographic codes based on standard classifications. See also **UNSPSC**.

Index

M

N

O

P

R

S

T

V

W