Oracle® Fusion Middleware

Administering Web Services

12*c* (12.2.1.4.0)
F23320-01
September 2019

Oracle Fusion Middleware Administering Web Services, 12*c* (12.2.1.4.0)

F23320-01

Primary Author: Oracle Corporation

Contents

4 Administering Web Services

5 Testing Web Services

List of Figures

List of Tables

Preface

This section describes the intended audience, how to use this guide, and provides information about documentation accessibility.

About this Guide

This guide describes the tasks required to administer Web services, providing details describing how to:

- Deploy, configure, test, and monitor Web services.
- Manage diagnostic and message logs.
- Manage policy lifecycle to transition from a test to production environment.
- Enable, publish, and register Web services.

Audience

This guide is intended for:

- System and security administrators who administer Web services and manage security
- Application developers who are developing Web services and testing the security prior to deployment of the Web services
- Security architects who create security policies

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following Oracle Web Services Manager documents:

- "Developing and Securing Web Services" in the *Developing Applications with Oracle JDeveloper*
- *Developing Extensible Applications for Oracle Web Services Manager*
- *Developing Fusion Web Applications with Oracle Application Development Framework*
- *Developing JAX-RPC Web Services for Oracle WebLogic Server*
- *Developing JAX-WS Web Services for Oracle WebLogic Server*
- *Developing Oracle Infrastructure Web Services*
- *Developing SOA Applications with Oracle SOA Suite*
- *Interoperability Solutions Guide for Oracle Web Services Manager*
- *Securing WebLogic Web Services for Oracle WebLogic Server*
- *Securing Web Services and Managing Policies with Oracle Web Services Manager*
- *Understanding Oracle Web Services Manager*
- *Understanding WebLogic Web Services for Oracle WebLogic Server*
- *Understanding Web Services*
- *Use Cases for Securing Web Services Using Oracle Web Services Manager*
- *WebLogic Web Services Reference for Oracle WebLogic Server*

See also the *Oracle Web Services Manager* Technology page at: `http://www.oracle.com/technetwork/middleware/webservices-manager/index.html`.

Conventions

The following text conventions are used in this document:

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

What's New in This Guide?

This preface introduces the new and changed features of Web Services management and provides pointers to additional information.

Topics:

New and Changed Features for 12*c* (12.2.1.4.0)

This revision contains no new features.

For a comprehensive listing of the new Oracle Web Services Manager features introduced in this release, see New and Changed Features for 12c (12.2.1.4.0) in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

New and Changed Features for 12*c* (12.2.1.3.0)

A new Virtual User Support for SAML/JWT has been provided which helps Mobile Cloud Service (MCS) to configure OWSM and validate identity tokens (SAML/JWT) from trusted issuers, and to populate the subject with role principals derived from token attributes/claims, according to configured rules.

See Configuring Virtual User Using WLST.

For a comprehensive listing of the new Oracle Web Services Manager features introduced in this release, see New and Changed Features for 12c (12.2.1.3.0) in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

New and Changed Features for 12*c* (12.2.1.2.0)

Minor updates, such as fixes or corrections, were made to this document.

For a comprehensive listing of the new Oracle Web Services Manager features introduced in this release, see New and Changed Features for 12c (12.2.1.2.0) in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

New and Changed Features for 12*c* (12.2.1.1.0)

Test Web Service page now includes additional information on custom policies.

See "Testing Security ".

For a comprehensive listing of the new Oracle Web Services Manager features introduced in this release, see New and Changed Features for 12c (12.2.1.1.0) in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

1

Overview of Web Services Administration

This chapter provides an overview of web services administration in Oracle Fusion Middleware 12*c*. Companies worldwide are actively deploying service-oriented architectures (SOA) using web services, both in intranet and internet environments. While web services offer many advantages over traditional alternatives (for example, distributed objects or custom software), deploying networks of interconnected web services still presents key challenges, particularly in terms of security and administration.
This chapter includes the following sections:

- Web Services Administration in Oracle Fusion Middleware 12*c*
- Roadmap for Web Service Administration Tasks

For definitions of unfamiliar terms found in this and other books, see the Glossary.

1.1 Web Services Administration in Oracle Fusion Middleware 12*c*

Oracle Fusion Middleware 12*c* supports Oracle Infrastructure web services and Java EE web services. At development time, application developers can secure web services by attaching policies using Oracle JDeveloper or programmatically using annotations. Post-deployment, system administrators can secure web services by attaching policies to the service endpoints or globally by reference to a range of endpoints.

In addition to securing the web services, system administrators may need to:

- Deploy, configure, test, and monitor web services.
- Enable, publish, and register web services.
- Manage policy lifecycle to transition from a test to production environment.
- Test interoperability with other web services.

For details about securing the web services and managing web services policies, see *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

1.2 Roadmap for Web Service Administration Tasks

Use the tools defined in the following table to secure and administer web services.

For information about how to access these administration tools, see Accessing the Administration Tools .

Table 1-1 Tools Used for Web Service Administration Tasks

Use this tool...	To perform the following tasks...
Oracle Enterprise Manager Fusion Middleware Control	• Deploy web service applications. • View and configure RESTful and Oracle Infrastructure web services and clients. • View Java EE (WebLogic) web services and clients. • Secure Java EE, RESTful, and Oracle Infrastructure web services with OWSM policies. For more information about securing web services using OWSM, see *Securing Web Services and Managing Policies with Oracle Web Services Manager*. • Test web services. • Monitor the run-time performance and audit web services. • Manage diagnostic and message logs. • Register web services and sources, and publish registered web services. **Note**: • Only a subset of OWSM policies are supported for Java EE web services, as described in "Which OWSM Policies are Supported for Java EE Web Services" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*. • Only a subset of OWSM policies are supported for RESTful web services, as described in "Which OWSM Policies are Supported for RESTful Web Services" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*. • Security and administration of JAX-RPC WebLogic web services is not supported.
WebLogic Scripting Tool (WLST)	• View and configure Oracle Infrastructure web services and clients. • View Java EE web services and clients. • Secure Oracle Infrastructure web services and Java EE web services with OWSM policies. For more information about securing web services using OWSM, see *Securing Web Services and Managing Policies with Oracle Web Services Manager*. • Manage application migration between environments, such as from test to production.
Oracle WebLogic Server Administration Console	Secure and manage Java EE web services. For more information about using the Oracle WebLogic Server Administration Console to secure and administer Java EE web services, see "Web Services" in the *Oracle WebLogic Server Administration Console Online Help*. **Note**: If available, Oracle Enterprise Manager Fusion Middleware Control is the preferred graphical user interface (GUI) tool for securing and managing Java EE web services.

2
Accessing the Administration Tools

This chapter introduces the several administration tools you can use to secure and administer Fusion Middleware Web Services. This chapter describes how to access the security and administration tools.
This chapter includes the following sections:

- Accessing Oracle Enterprise Manager Fusion Middleware Control
- Accessing Oracle WebLogic Administration Console
- Accessing the Web Services Custom WLST Commands

2.1 Accessing Oracle Enterprise Manager Fusion Middleware Control

Use the Oracle Enterprise Manager Fusion Middleware Control to manage services in your enterprise, including hosts, databases, listeners, application servers, HTTP Servers, and Web applications as one cohesive unit.
To access Oracle Enterprise Manager Fusion Middleware Control:

1. Start the Oracle WebLogic Server instance.

   For more information, see "Start and stop servers" in the *Oracle WebLogic Server Administration Console Online Help*.

2. Open a supported web browser and navigate to the following URL:

   ```
   http://hostname:port/em
   https://hostname:port/em
   ```

   hostname specifies the DNS name or IP address of Fusion Middleware Control and *port* specifies the address of the port on which Fusion Middleware Control is listening for requests (7001 by default).

   Use `https` if you started the Oracle Fusion Middleware using the Secure Sockets Layer (SSL).

   For a list of supported browsers, see *Oracle Fusion Middleware System Requirements and Specifications* at: `http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html`

   The Login page displays.

3. Enter the username and password.

   The default user name for the administrator user is `weblogic`. This is the account you can use to log in to Fusion Middleware Control for the first time. The password is the one you supplied during the installation of Oracle Fusion Middleware.

4. Click **Login**.

> **Note:**
>
> The tasks and functions available for managing web services using Fusion Middleware Control depend on your role. For information, see Understanding Access Privileges for the Supported User Roles.

For more information, see "Getting Started Using Oracle Enterprise Manager Fusion Middleware Control" in *Administering Oracle Fusion Middleware*.

2.2 Accessing Oracle WebLogic Administration Console

You can access the WebLogic System Administration Console to configure security parameters, including managing users, groups, and roles.
To access Oracle WebLogic Administration Console:

1. Start the Oracle WebLogic Server.

   For more information, see "Start and stop servers" in the *Oracle WebLogic Server Administration Console Online Help*.

2. Open a supported web browser and navigate to one of the following URLs:

   ```
   http://hostname:port/console
   https://hostname:port/console
   ```

   hostname specifies the DNS name or IP address of the Oracle WebLogic Administration Server and *port* specifies the address of the port on which the Oracle WebLogic Administration Server is listening for requests (7001 by default).

   Use `https` if you started the Oracle WebLogic Server using the Secure Sockets Layer (SSL).

   For a list of supported browsers, see *Oracle Fusion Middleware System Requirements and Specifications* at: `http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html`

   The Login page displays.

3. Enter the username and password.

   You may have specified the username and password during the installation process. This may be the same username and password that you use to start the Oracle Administration Server. Or, a username that is granted one of the default global security roles.

4. Click **Log In**.

For more information, see "Start the Console" in the *Oracle WebLogic Server Administration Console Online Help*.

2.3 Accessing the Web Services Custom WLST Commands

The WebLogic Scripting Tool (WLST) is a command-line scripting environment that you can use to create, manage, and monitor WebLogic domains.
To access the web services WLST commands:

1. Go to the Oracle Common home directory for your installation, for example `/home/Oracle/Middleware/oracle_common`.

For information about the Oracle Common home directory and installing Oracle Fusion Middleware, see the *Planning an Installation of Oracle Fusion Middleware*.

2. Start WLST using the `WLST.sh/cmd` command located in the `oracle_common/common/bin` directory. For example:

   - `/home/Oracle/Middleware/oracle_common/common/bin/wlst.sh` (UNIX)
   - `C:\Oracle\Middleware\oracle_common\common\bin\wlst.cmd` (Windows)

   When executed, these commands start WLST in offline mode. To use the web services WLST commands, you must use WLST in online mode.

3. Start Oracle WebLogic Server.

   For more information, see "Start and stop servers" in the *Oracle WebLogic Server Administration Console Online Help*.

4. Connect to the running WebLogic Server instance using the `connect()` command. For example, the following command connects WLST to the Admin Server at the URL `myAdminServer.example.com:7001` using the username/password credentials `weblogic/password`:

   ```
   connect("weblogic","password","t3://myAdminServer.example.com:7001")
   ```

For more information about using WLST, see "Using the WebLogic Scripting Tool" in *Understanding the WebLogic Scripting Tool*.

For more information about the web Services WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

3
Deploying Web Service Applications

As you work with web services, you will find that you can deploy and undeploy the associated applications in different ways.
Follow these guidelines when deploying applications associated with web services:

- Use Oracle Enterprise Manager Fusion Middleware Control to deploy Java EE applications that require Oracle Metadata Services (MDS) or that take advantage of the Oracle Application Development Framework (Oracle ADF). For more information, see *Deploying, Undeploying, and Redeploying Java EE Applications.*
- If your application is a SOA composite, use the SOA Composite deployment wizard. For more information, see *Deploying, Undeploying, and Redeploying SOA Composite Applications.*
- If your application is not a SOA composite or it does not require an MDS repository or ADF connections, then you can deploy your application using Fusion Middleware Control or the Oracle WebLogic Server Administration Console. For more information, see *Deploying, Undeploying, and Redeploying Oracle ADF Applications.*

For more information about deploying applications, see "Deploying Applications" in *Administering Oracle Fusion Middleware*.

This chapter includes the following sections:

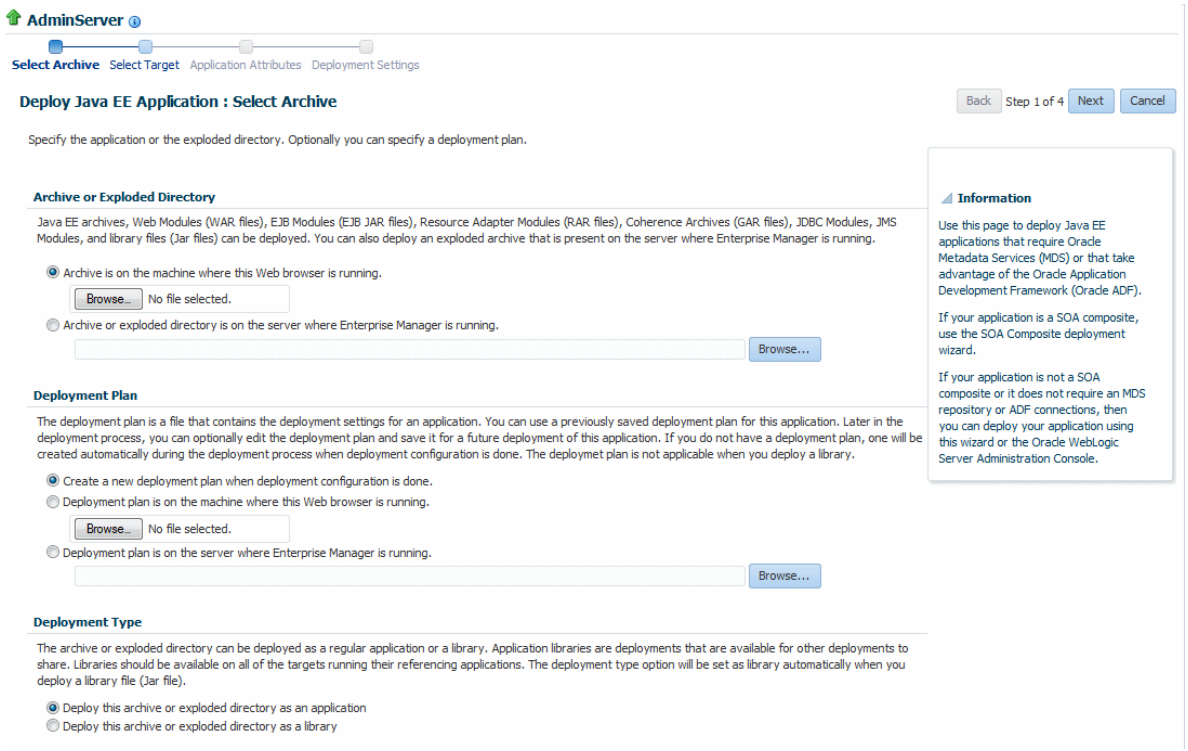
3.1 Deploying Web Service Applications

The following is an overview of the basic procedure for deploying a web service application using the Oracle Enterprise Manager Fusion Middleware Control.
To deploy a web service application:

1. In the navigation pane, expand **WebLogic Domain**.
2. Expand the domain in which you want to deploy the web service, and then select the instance of the server on which you want to deploy it.
3. In the content pane, select **WebLogic Server** then **Deployments**.
4. On the Deployments page, click **Deploy**.

   The first screen of the Deploy process is displayed, as shown in Figure 3-1.

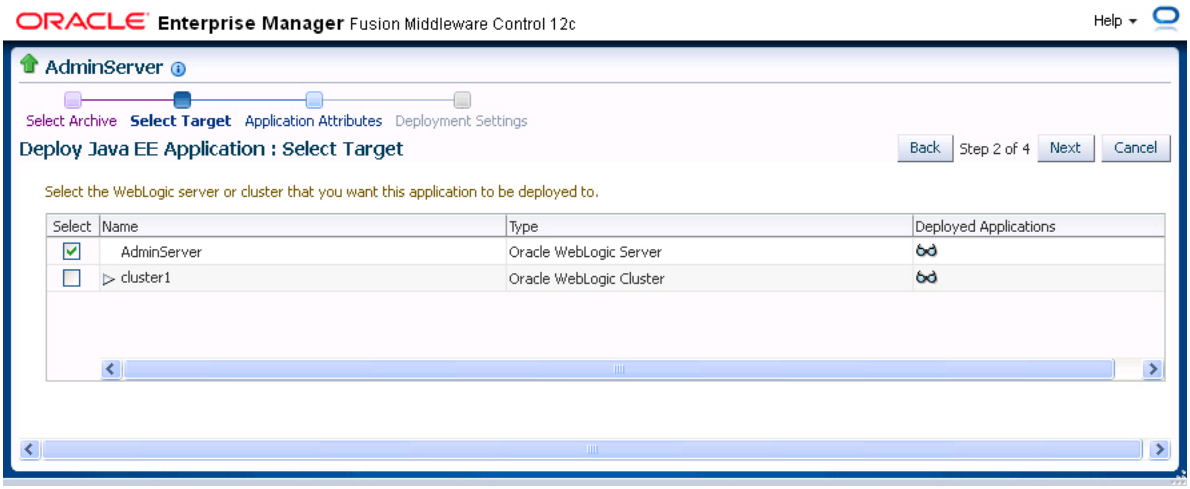
Figure 3-1 Select Archive Page



5. Select one of the following Archive or Exploded Directory options:
   - Archive is on the machine where this web browser is running.
   - Archive or exploded directory is on the server where Enterprise Manager is running.
6. Select one of the following Deployment Plan options:
   - Automatically create a new deployment plan
   - Deployment plan is present on local host
   - Deployment plan is already present on the server where Enterprise Manager is running

   A deployment plan is an XML file that you use to configure an application for deployment to a specific environment. If you do not already have a deployment plan for the web service application you are deploying, one is created for you when you deploy the application.
7. Select one of the following Deployment Type options:
   - Deploy this archive or exploded directory as an application
   - Deploy this archive or exploded directory as a library
8. Click **Next**.
9. On the Select Target page, select the target (WebLogic server or cluster) to which you want this application deployed, and click **Next**.

Figure 3-2 Select Target Page

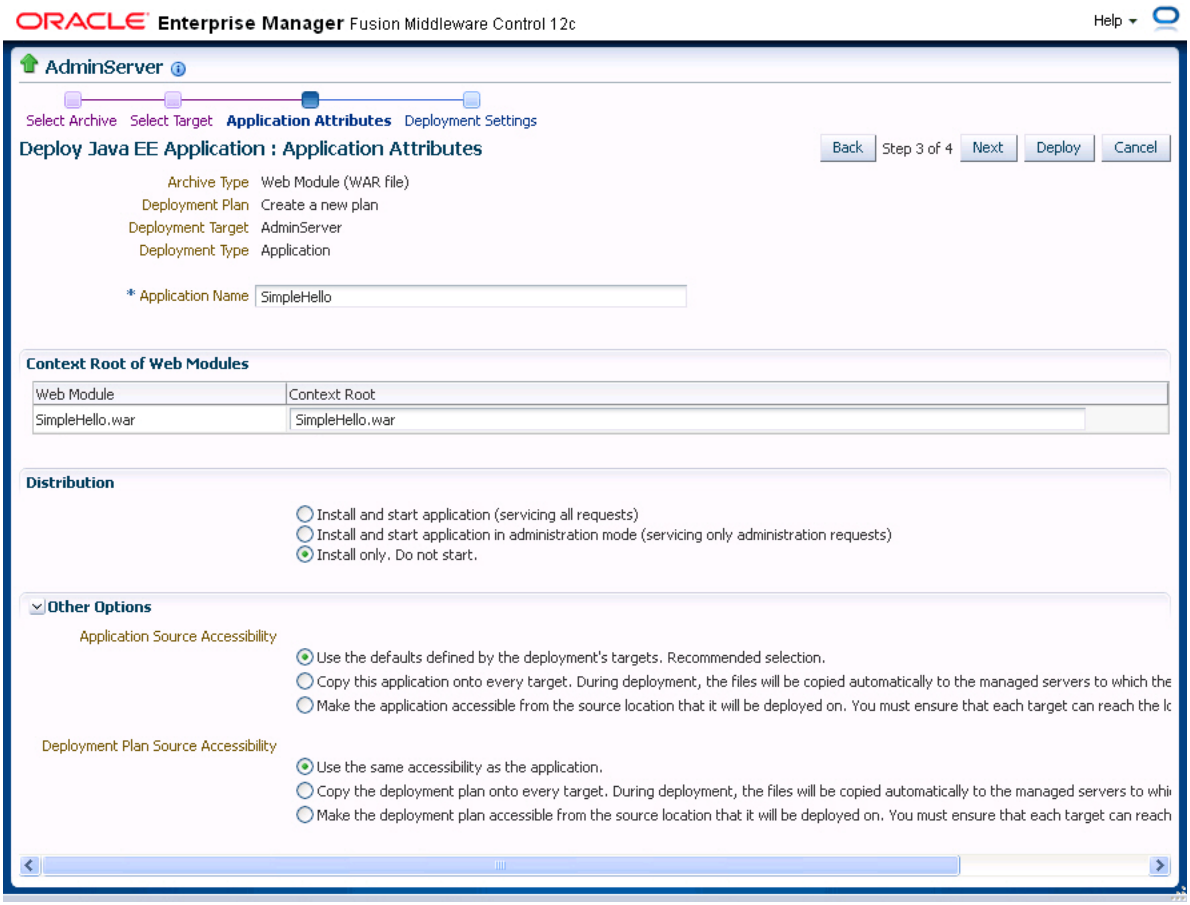


10. On the Application Attributes page, enter the attributes for this web service application, and click **Next.** Application Name is the only required attribute.

    However, if you want to be able to redeploy this web service application later without first having to undeploy it, you must also assign a version number.

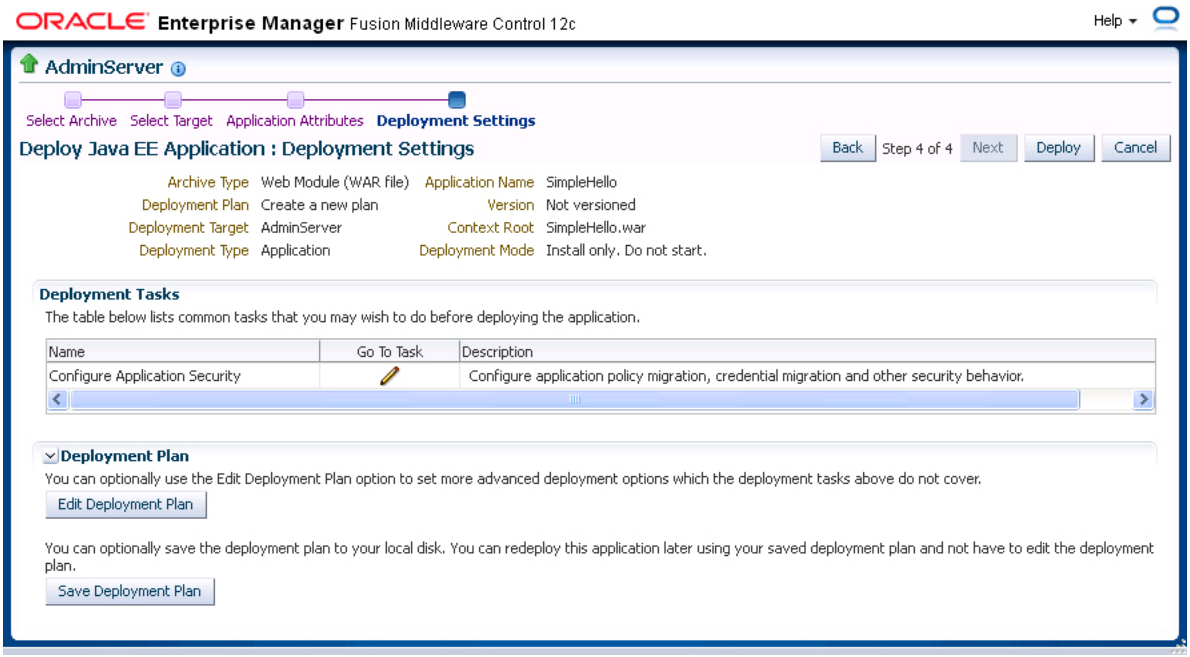
    The context root is the URI for the web module. Each web module or EJB module that contains web services may have a context root.

Figure 3-3 Application Attributes Page



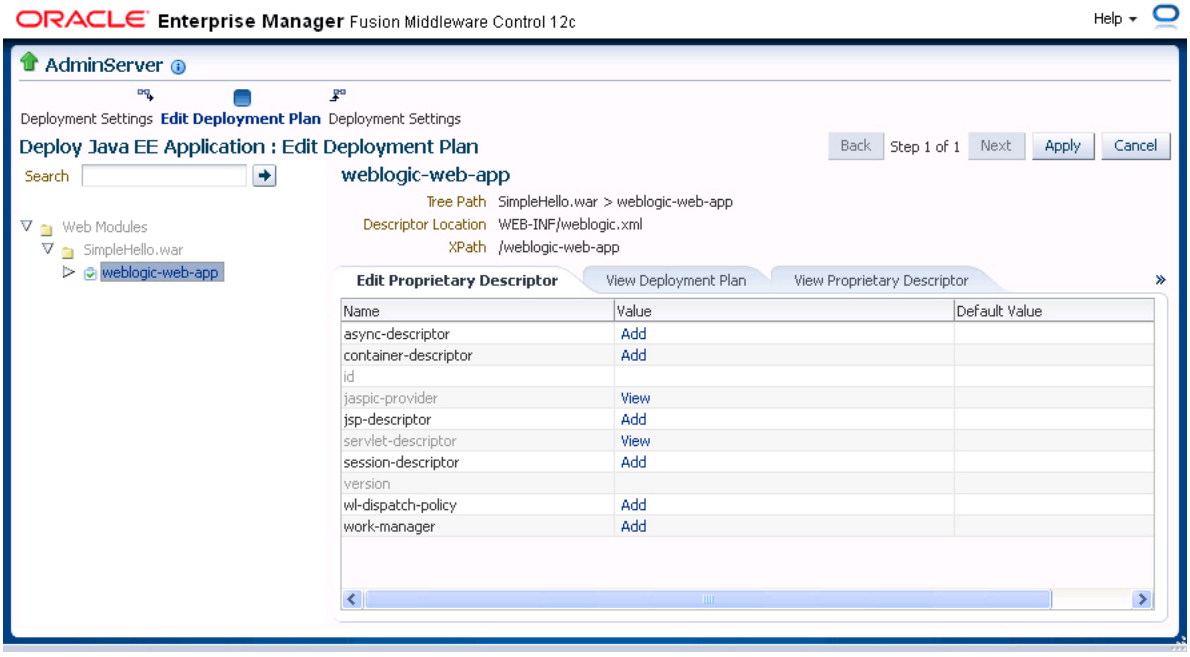
11. On the Deployment Settings page, edit the deployment settings for this web service application, as shown in Figure 3-4.

Figure 3-4 Deployment Settings Page



12. To save a copy of the deployment plan to your local system, click **Save Deployment Plan**.
13. To edit the deployment plan, possibly to add advanced deployment options, click **Edit Deployment Plan**. If you do so, the Edit Deployment Plan screen is displayed, as shown in Figure 3-5. After making changes to the deployment plan, click **Apply** to make the change effective.

Figure 3-5 Edit Deployment Plan



14. Click **Deploy** on the Deployment Settings page. If successful, the Deployment Succeeded screen is displayed.

3.2 Undeploying Web Service Applications

The procedure for undeploying or redeploying a web service is the same as the procedure for any application.
To undeploy a web service application:

1. From the navigation pane, expand **Application Deployments**, then expand the application deployment and select the application name

   The Application Deployment is displayed

2. In the content pane, select **Application Deployment** then **Deployments**.
3. In the Deployments table, select the application and click **Undeploy**.

   The undeploy confirmation page is displayed.

4. Click **Undeploy**.

   Processing messages are displayed.

5. When the operation completes, click **Close**.

3.3 Redeploying Web Service Applications

When you redeploy a web service application, the running application is automatically stopped and then restarted.
Redeploy an application if:

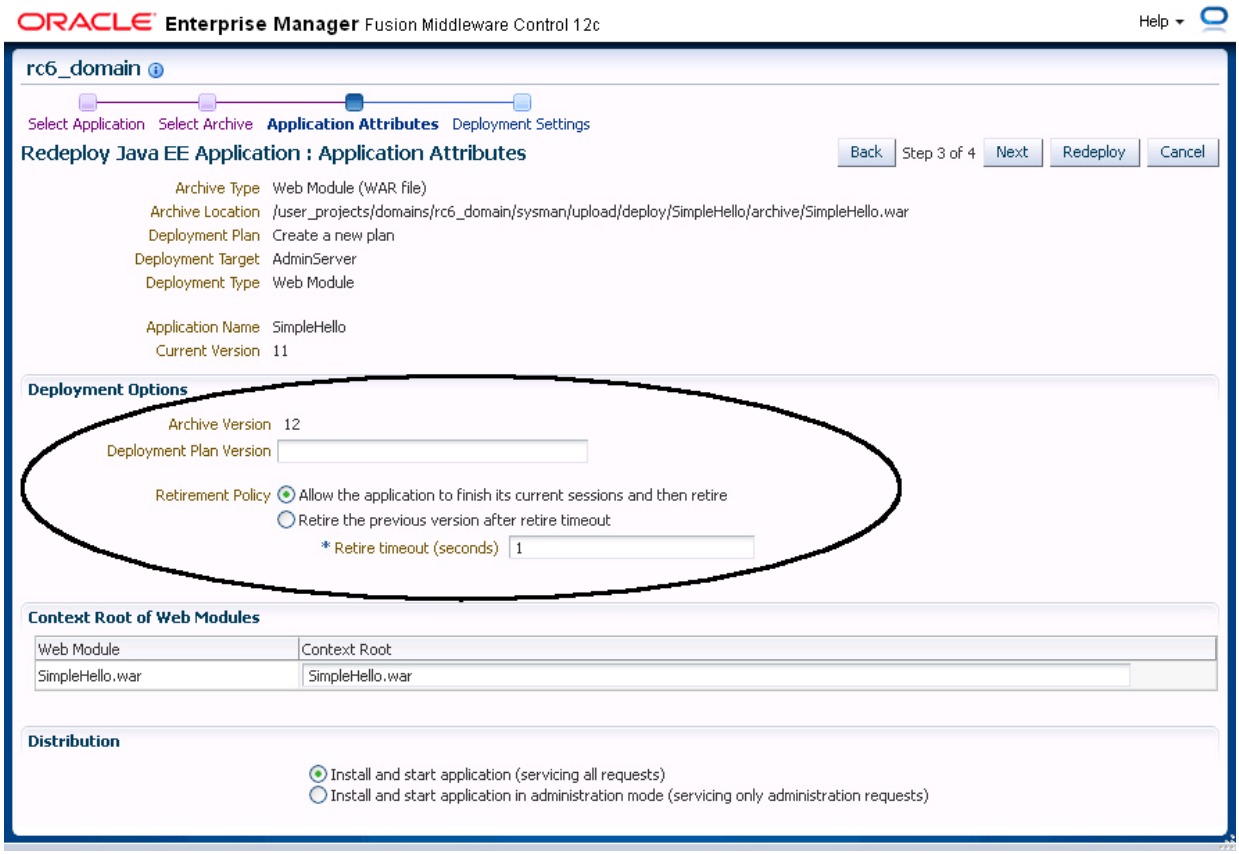
- You have made changes to the application and you want to make the changes available.
- You have made changes to the deployment plan.
- You want to redeploy an entirely new archive file in a new location.

When you redeploy an application, you can redeploy the original archive file or exploded directory, or you can specify a new archive file in place of the original one. You can also change the deployment plan that is associated with the application.

> **Note:**
>
> Applications that were previously deployed without a version cannot be redeployed. To redeploy the not-versioned applications, you need to undeploy and deploy the application.

The steps that you follow to redeploy a web service application are identical to those required when you first deployed the application, as described in "Deploying Web Service Applications", with two exceptions: you must redeploy the application with a new version, and you can optionally set the retirement policy for the current version. Both of these actions occur at Step 3 of redeployment process, as shown in Figure 3-6.

Figure 3-6 Setting Application Attributes During Redeploy



4
Administering Web Services

This chapter describes how to use Oracle Enterprise Manager Fusion Middleware Control and WebLogic Scripting Tool (WLST) commands to configure and manage Oracle Fusion Middleware Web Services.
This chapter includes the following sections:

- Overview of Web Services Administration Using Fusion Middleware Control
- Overview of Web Services Administration Using WLST

4.1 Overview of Web Services Administration Using Fusion Middleware Control

You can use Fusion Middleware Control to access and configure web services on the server and client.

- Understanding Access Privileges for the Supported User Roles
- Introduction to Viewing Web Services Using Fusion Middleware Control
- Introduction to Viewing Web Service Clients Using Fusion Middleware Control
- Introduction to Configuring Web Services Using Fusion Middleware Control
- Overview of Configuring Web Service Clients Using Fusion Middleware Control
- Overview of Managing the WSDL Using Fusion Middleware Control

4.1.1 Understanding Access Privileges for the Supported User Roles

Fusion Middleware Control supports the notion of role-based access. Users are mapped to different roles, and each role corresponds to a different set of privileges. Fusion Middleware Control uses the Oracle WebLogic Server security realm and the roles defined in that realm.
Web service management tasks pertaining to assertion templates and policies can be restricted by role, as shown in Table 4-1.

Table 4-1 Oracle Web Services Manager Privileges for Supported Roles

Privileges	Administrator	Deployer	Monitor	Operator
Read policies, assertion templates, and policy sets.	Yes	Yes	Yes	Yes

Table 4-1 (Cont.) Oracle Web Services Manager Privileges for Supported Roles

Privileges	Administrator	Deployer	Monitor	Operator
Create, update, and delete policies and assertion templates. **Note:** None of the roles provide permission to update or delete the *predefined* policies and assertion templates, which are read-only. You must *clone* a predefined policy or assertion template before modifying it; and you can clone policies in the security and management categories only. For more information, see "Managing Web Service Policies with Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.	Yes	No	No	No
Create, update, and delete policy sets.	Yes	No	No	No
Attach and detach OWSM policies.	Yes	No	No	No
Generate client policies.	Yes	No	No	No
Check compatibility of service policies and client policies.	Yes	Yes	Yes	Yes

For more information about users and roles in Fusion Middleware Control, see "Understanding Users and Roles for Fusion Middleware Control" in *Administering Oracle Fusion Middleware*.

4.1.2 Introduction to Viewing Web Services Using Fusion Middleware Control

You can use the Fusion Middleware Control to view web services on a server, and in an application deployment or application.

- Viewing the Web Services for a Server Using Fusion Middleware Control
- Viewing the Web Services in an Application Deployment Using Fusion Middleware Control
- Viewing the Web Services Summary Page for an Application
- Viewing the Summary Page for a Java EE Web Service
- Viewing the Web Services and References in a SOA Composite
- Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control

4.1.2.1 Viewing the Web Services for a Server Using Fusion Middleware Control

Follow the procedures below to view all of the currently deployed web services for a given server.

To view the web services for a server:

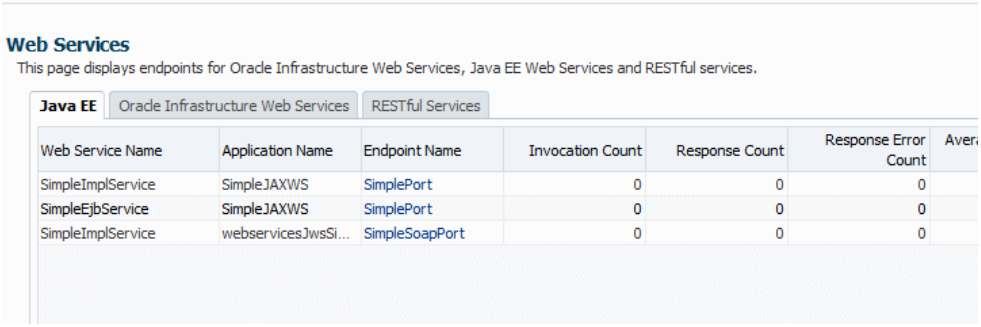
1. In the navigation pane, expand **WebLogic Domain** to show the domains.
2. Expand the domain name to see the list of servers.

3. Select the server for which you want to view all current web services.
4. In the content pane, select **WebLogic Server** and then **Web Services**. The Web Services server summary page displays, as shown in Figure 4-1.

   You can view tabs for Java EE web services, Oracle Infrastructure web services, such as ADF and RESTful services.

   The tabs that are displayed depend on the web services deployed on that server.

Figure 4-1 Web Services Summary Page for a Server



4.1.2.2 Viewing the Web Services in an Application Deployment Using Fusion Middleware Control

Fusion Middleware Control allows you to view a summary of the currently deployed web services for a given application deployment. The summary appears on the Application Deployment home page.

To view the web services in an application deployment:

1. In the navigation pane, expand the **Application Deployments** folder to view the application deployments in the domain.
2. Select the application deployment.

The Web Services summary appears on the right side of the Domain Application Deployment page. For each service, the summary lists the name of the Web Service, the name of server it is running on, the name of the associated application running on the server, the name of the web service endpoint, and a link to a test page for the service.

> **Note:**
>
> The Web Services summary does not include RESTful web services.

4.1.2.3 Viewing the Web Services Summary Page for an Application

Follow the procedure below to navigate to the page where you can see the list of web services for your application.

To view the web services summary page for an application:

1. In the navigation pane, expand the **Application Deployments** folder to expose the application deployments in the domain and select the domain application deployment target.

   The Domain Application Deployment home page is displayed in the content pane.

2. In the navigation pane, expand the selected application deployment (and any of its child nodes such as cluster application deployments, if applicable, to expose application deployments) and select the application deployment target.

   The Application Deployment home page is displayed in the content pane.

3. In the content pane, select **Application Deployment**, then **Web Services**.

   The Web Services application summary page is displayed.

   From the Web Services application summary page, you can do the following:

   - View the web services in the application.
   - View the web service configuration, endpoint status, policy faults, and more. (ADF applications only.)
   - View and monitor web services faults, including Security, Reliable Messaging, MTOM, Management, and Service faults. (ADF applications only.)
   - View and monitor Security violations, including authentication, authorization, message integrity, and message confidentiality violations. (ADF applications only.)
   - Navigate to pages where you can configure your web services endpoints, including enabling and disabling the endpoint, and attaching policies to web services.

   The Web Service Details table contains tabs where you can view more information about web services and ports, web service endpoints, RESTful services, and Java EE web service clients.

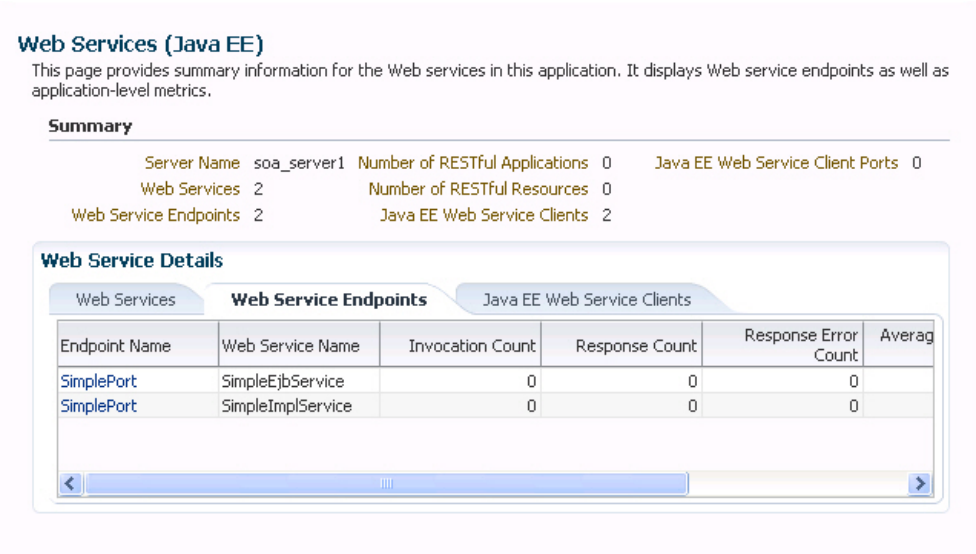
   Figure 4-2 shows the Web Services application summary page for an ADF application.

> **Note:**
>
> The RESTful Services and Java EE Web Service Clients tabs are displayed only if there are RESTful services or Java EE Web Service client instances, respectively, in the application.

Figure 4-2 Web Services Summary Page for ADF Applications

Figure 4-3 shows the Web Services summary page for a Java EE application. The Web Services (Java EE) page displays information about the web services in this application, such as the numbers of web services, web service endpoints, web service clients and client ports, and the number of RESTful applications and resources.

Figure 4-3 Web Services Summary Page for Java EE Applications



4.1.2.4 Viewing the Summary Page for a Java EE Web Service

Follow this procedure to view summary information for a Java EE Web Service.

1. Navigate to the Web Services summary page for the Java EE application, as described in "Viewing the Web Services Summary Page for an Application".
2. In the Web Services tab, click the name of the web service for which you want to view summary information.

The Java EE Web Service summary page displays two tabs: **Web Service Endpoints** and **Invocations**. The **Web Service Endpoints** tab displays the endpoints associated with the web service. Invocation statistics are aggregated data for the Java EE web service. For more information on these statistics, see "Viewing Operation Statistics for a Web Service Endpoint".

The **Invocations** tab provides statistics on the web service invocations, such as the invocation and response error count. For more information on these statistics, see "Viewing Statistics for a Java EE Web Service Operation".

4.1.2.5 Viewing the Web Services and References in a SOA Composite

Use the following procedure to view the SOAP or RESTful web services, references, and components in a SOA composite application:

1. In the navigation pane, expand the **SOA** folder.
2. Expand **soa-infra** to view the SOA partitions, then expand the SOA partition (for example, the default partition) and select the target SOA composite application.

   The SOA composite home page displays.
3. Select the **Dashboard** tab if it is not already selected.

   The Components section of this tab lists the SOA components being used in the composite application, and the Services and References section displays the web service and reference bindings.

4.1.2.6 Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control

In Fusion Middleware Control, the following sections describe the steps you follow to view the details for a web service endpoint:

- Viewing the Details for a SOA Composite Application
- Viewing the Details for a non-SOA Oracle Infrastructure Web Service Endpoint
- Viewing the Details for a Java EE Web Service Endpoint
- Viewing the Details for a Java EE Web Service Operation
- Viewing the Details for a RESTful Service Application
- Viewing the Details for a RESTful Resource

4.1.2.6.1 Viewing the Details for a SOA Composite Application

Follow this procedure to view the web service endpoint configuration for a SOA composite application.

1. Navigate to the home page for the SOA composite as described in "Viewing the Web Services and References in a SOA Composite".
2. In the Services and References section of the page, click the name of the service or reference to display the Service Home or Reference Home page, as appropriate.
3. From the Service Home or Reference Home page, you can do the following:
   - Click the **Dashboard** tab, if it is not already selected, to see a graphic representation of the total incoming messages and faults since server startup, and recently rejected messages, including the message name, time of the fault, and the type of fault (business or system).
   - Click the **Policies** tab to view or change the policies attached to this endpoint.
   - Click the **Properties** tab to view and modify the configuration for this endpoint.

     > **Note:**
     >
     > The **Properties** tab is not available for RESTful SOA component services.

   - Click the **Adapter Reports** tab to view and enable a list of diagnosability reports. This tab is only displayed for adapter endpoints.

   For additional information about SOA composite endpoints, see "Administering Binding Components" in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

4.1.2.6.2 Viewing the Details for a non-SOA Oracle Infrastructure Web Service Endpoint

Follow this procedure to view the details for a non-SOA Oracle Infrastructure web service endpoint (such as ADF).

1. Navigate to the Web Services Application Summary page as described in "Viewing the Web Services Summary Page for an Application".
2. In the Web Service Details section of the page, click the **Web Services** tab and expand the web service to display the web service endpoints if they are not already displayed.

   Alternatively, click the **Web Service Endpoints** tab to view a list of web service endpoints.
3. Click the name of the endpoint to navigate to the Web Service Endpoint page.
4. From the Web Service Endpoint page, you can do the following:
   - Click the **Operations** tab to see the list of operations for this endpoint.
   - Click the **WSM Policies** tab to see the policies attached to this endpoint, if the endpoint has a valid configuration, and if it is secure.
   - Click the **Charts** tab to view a graphical view of the faults for this endpoint. There are two charts, one that shows the distribution of all faults, and a second that shows the distribution of only security faults. For more information, see "Overview of Viewing the Security Violations for a Web Service".
   - Click the **Configuration** tab to configure properties for the web service endpoint. For more information, see "Introduction to Configuring Web Services Using Fusion Middleware Control".

As an alternative method of viewing the details for a web service endpoint, you can instead navigate to the Web Services Server Summary page, as described in "Viewing the Web Services for a Server Using Fusion Middleware Control", which lists all of the web services, and click the name of the endpoint to navigate to the specific Web Service Endpoint page.

4.1.2.6.3 Viewing the Details for a Java EE Web Service Endpoint

Follow this procedure to view the details for a Java EE web service endpoint.

1. Navigate to the Web Services Summary page as described in "Viewing the Web Services Summary Page for an Application".
2. In the Web Service Details section of the page, click the **Web Services** tab and expand the web service to display the web service endpoints if they are not already displayed.

   Alternatively, click the **Web Service Endpoints** tab to view a list of web service endpoints.
3. Click the name of the endpoint to navigate to the Web Service Endpoint page.
4. From the Web Service Endpoint page, you can do the following:
   - Click the **Operations** tab to see the list of operations for this endpoint.
   - Click the **Invocations** tab to see the error, invocation, and response statistics associated with invoking the endpoint.
   - Click the **WSM Policies** tab to see the policies attached to this endpoint, if the endpoint has a valid configuration, and if it is secure.

> **Note:**
>
> You can also view details about security violations for an endpoint. For more information, see "Overview of Viewing the Security Violations for a Web Service".

As an alternative method of viewing the details for a web service endpoint, you can instead navigate to the server-wide Web Services Summary page, as described in "Viewing the Web Services for a Server Using Fusion Middleware Control", which lists all of the web services, and click the name of the endpoint to navigate to the specific Web Service Endpoint page.

4.1.2.6.4 Viewing the Details for a Java EE Web Service Operation

Follow this procedure to view the operations belonging to a Java EE web service.

1. Navigate to the summary page for the Java EE web service as described in "Viewing the Summary Page for a Java EE Web Service".
2. In the Web Services tab, select the name of the endpoint for which you want to view the details.
3. Click the **Operations** tab of the Web Service Endpoint page to display the operations associated with the endpoint.
4. Click the name of the operation to display more detailed information.

The Web Service Operation page displays information about the operation such as the endpoint URI, the names of the application, web service, and endpoint to which it is associated, and error, invocation, and response statistics.

For descriptions of the endpoint details and the statistics, see "Viewing Statistics for a Java EE Web Service Operation".

4.1.2.6.5 Viewing the Details for a RESTful Service Application

Follow this procedure to view the details of a RESTful service application.

1. Navigate to the Web Services Application Summary page as described in "Viewing the Web Services Summary Page for an Application".
2. In the Web Service Details section of the page, click the **RESTful Services** tab.
3. Click the name of the RESTful application to navigate to the RESTful Service Application page.
4. From the RESTful Service Application page, you can do the following:
   - Click the **RESTful Resources** tab to view the list of resources associated with the RESTful service. For each resource, you can view its name, type, path, invocation count, and average execution time in milliseconds.
   - Click the **WSM Policies** tab to see the policies attached to this RESTful service, if the RESTful service has a valid configuration, and if it is secure.

4.1.2.6.5.1 Viewing the Details for a RESTful Resource

Follow this procedure to view the details of a RESTful resource.

1. Navigate to the Web Services Application Summary page as described in "Viewing the Web Services Summary Page for an Application".
2. In the Web Service Details section of the page, click the **RESTful Services** tab.
3. Click the name of the RESTful application to navigate to the RESTful Service Application page.
4. From the RESTful Service Application page, click the **RESTful Resources** tab to see the list of resources associated with the RESTful service.
5. Click the name of the RESTful resource for which you want to view more information.

   The RESTful Resource page displays information about the resource such as the application, module, and RESTful application name; the resource name, type, and path; the number of methods and suppressors locators; and statistics including invocation count and average execution time in milliseconds.

   The RESTful Methods tab displays the information about the resource methods, including the method name, return type, path, associated HTTP method, and media type produced. Statistics including invocation count, average execution time, and execution time totals are also displayed.

4.1.3 Introduction to Viewing Web Service Clients Using Fusion Middleware Control

You can use the Fusion Middleware Control to view web service clients depending on the application type (SOA reference, ADF DC or asynchronous Callback client). The steps you follow to view a web service client depends on the application type you use as described in the following sections:

- Viewing SOA References
- Viewing Connection-Based Web Service Clients
- Viewing Java EE Web Service Clients
- Viewing Asynchronous Web Service Callback Clients

4.1.3.1 Viewing SOA References

Use the following procedure to view a SOA reference client:

1. In the navigation pane, expand the **SOA** folder.
2. Expand **soa-infra**, then expand the SOA partition (for example, the default partition) and select the target SOA composite application.

   The SOA composite home page displays.
3. Click the **Dashboard** tab, if it is not already selected.
4. In the Services and References portion of the page, select the SOA reference to view.
5. From the Reference Home page, you can do the following:
   - Click the **Dashboard** tab, if it is not already selected, to see a graphic representation of the total incoming messages and faults since server startup, and recently rejected messages, including the message name, time of the fault, and the type of fault (business or system).

- Click the **Policies** tab to view or change the policies attached to this endpoint.
- Click the **Properties** tab to view and modify the configuration for this endpoint.
- Click the **Adapter Reports** tab to view and enable a list of diagnosability reports. This tab is only displayed for adapter endpoints.

For additional information about SOA composite endpoints, see "Administering Binding Components" in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

4.1.3.2 Viewing Connection-Based Web Service Clients

Use the following procedure to view a connection-based web service client such as an ADF DC web service client or ADF JAX-WS Indirection Proxy:

1. In the navigation pane, expand the **Application Deployments** folder to expose the application deployments in the domain and select the domain application deployment target.

   The Domain Application Deployment home page is displayed in the content pane.
2. In the navigation pane, expand the selected application deployment (and any of its child nodes such as cluster application deployments, if applicable, to expose application deployments) and select the application deployment target.

   The Application Deployment home page is displayed in the content pane.
3. From the **Application Deployment** menu, select **ADF**, and then **Configure ADF Connections**.
4. On the ADF Connections Configuration page, select a connection from the Web Service Connections section of the page, and then select the endpoint from the **Advanced Connection Configuration** list.
5. In the Configure Web Service page, click the tabs to view the client data.

4.1.3.3 Viewing Java EE Web Service Clients

Use the following procedure to view Java EE web service clients:

1. In the navigation pane, expand the **Application Deployments** folder to expose the applications in the domain and select the domain application deployment target.

   The Domain Application Deployment home page is displayed in the content pane.
2. In the navigation pane, expand the selected application deployment (and any of its child nodes such as cluster application deployments, if applicable, to expose application deployments) and select the Java EE application deployment target.

   The Application Deployment home page is displayed in the content pane.
3. From the **Application Deployment** menu, select **Web Services**.

   The Web Services (Java EE) summary page is displayed.
4. Select the **Java EE Web Service Clients** tab to view the clients in the application.
   - Use the **Monitoring** tab to view the run-time client instances in the application. For more information, see "Viewing Statistics for Java EE Web Service Clients".

- Use the **Configuration** tab to view the client ports and attach or detach policies. For more information, see "Attaching Policies Directly to Web Service Clients" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.3.4 Viewing Asynchronous Web Service Callback Clients

Use the following procedure to view an asynchronous web service callback client. Callback clients are used only by asynchronous web services to return the response to the caller. For more information, see "Developing Asynchronous Web Services" in *Developing Oracle Infrastructure Web Services*.

1. Navigate to the endpoint for the asynchronous web service, as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".
2. Click **Callback Client** in the upper right portion of the endpoint page.

4.1.4 Introduction to Configuring Web Services Using Fusion Middleware Control

You can use the Fusion Middleware Control to configure web services.
The following sections detail the various procedure to configure web services.

- Configuring Addressing Using Fusion Middleware Control
- Configuring Asynchronous Web Services Using Fusion Middleware Control
- Changing the JMS System User for Asynchronous Web Services Using Fusion Middleware Control
- Configuring Reliable Messaging Using Fusion Middleware Control
- Configuring Atomic Transactions Using Fusion Middleware Control
- Configuring MTOM Using Fusion Middleware Control
- Configuring Fast Infoset Using Fusion Middleware Control
- Configuring SOAP Over JMS Transport Using Fusion Middleware Control
- Configuring Persistence Using Fusion Middleware Control
- Enabling or Disabling Web Services Using Fusion Middleware Control
- Enabling or Disabling Public Access to the Web Service WSDL Document Using Fusion Middleware Control
- Enabling or Disabling SOAP Processing Using Fusion Middleware Control
- Enabling or Disabling Non-SOAP XML Message Processing Using Fusion Middleware Control
- Setting the Log Level for Diagnostic Logs Using Fusion Middleware Control
- Introduction to Enabling or Disabling the Web Services Test Client Using Fusion Middleware Control
- Enabling or Disabling the Exchange of Metadata Using Fusion Middleware Control
- Configuring MTOM-encoded Fault Messages Using Fusion Middleware Control
- Validating the Request Message Using Fusion Middleware Control

- Setting the Size of the Request Message Using Fusion Middleware Control
- Enabling or Disabling Binary Content Caching Using Fusion Middleware Control

4.1.4.1 Configuring Addressing Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

To enable addressing using Fusion Middleware Control, attach the `oracle/wsaddr_policy` policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the addressing policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Attach the `oracle/no_addressing_policy` to disable an addressing policy configured at a higher scope.

For more information about the addressing policies, see "Addressing Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.4.2 Configuring Asynchronous Web Services Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to *asynchronous* non-SOA Oracle Infrastructure web services only.

When you invoke a web service synchronously, the invoking client application waits for the response to return before it can continue with its work. In cases where the response returns immediately, this method of invoking the web service might be adequate. However, because request processing can be delayed, it is often useful for the client application to continue its work and handle the response later on. By calling a web service asynchronously, the client can continue its processing, without interrupt, and will be notified when the asynchronous response is returned.

To configure asynchronous web services using Fusion Middleware Control, use one of the following methods:

- Attach the `oracle/async_web_service_policy` configuration policy to the web service, as described in "Attaching Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the asynchronous policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Attach the `oracle/no_async_web_service_policy` to disable an asynchronous policy configured at a higher scope.

  For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Use the Configuration tab, as described below.

You can also configure asynchronous *Callback client*, as described in "Configuring Asynchronous Web Service Callback Clients".

For information about developing asynchronous web services, see "Developing Asynchronous Web Services" in *Developing Oracle Infrastructure Web Services*.

To configure asynchronous web services using the Configuration tab:

1. Navigate to the Web Services Application Summary page.
2. In the Web Service Details section of the page, expand the web service to display the web service endpoints if they are not already displayed.
3. Click the name of the endpoint of the asynchronous web service to navigate to the Web Service Endpoint page.

   For an asynchronous web service, the Asynchronous flag at the top of the page is set to True. Review the following flags, which provide more information about the asynchronous web service:

   - Transaction Enabled for Request Queue—Flag that specifies whether transactions are enabled on the request queue.
   - Using Response Queue—Flag that specifies whether a response queue is being used. If set to false, then the response is sent directly to the web service client, without being stored.
   - Transaction Enabled for Response Queue—Flag that specifies whether transactions are enabled on the response queue.

   These flags are configured at design time. For more information, see "Developing Asynchronous Web Services" in *Developing Oracle Infrastructure Web Services*.
4. From the Web Service Endpoint page, click the **Configuration** tab.
5. Under the Asynchronous Web Service section of the page, you can set the configuration properties defined in Table 4-2.

> **Note:**
>
> The configuration properties defined in Table 4-2 appear and are valid only for asynchronous web services.

Table 4-2 Configuration Properties for Asynchronous Web Services

Configuration Property	Description
JMS Request Queue Connection Factory Name	Name of the connection factory for the JMS request queue. The default JMS connection factory, `weblogic.jms.XAConnectionFactory`, provided with the base domain is used by default.
JMS Request Queue Name	Name of the request queue. The following queue is used by default: `oracle.j2ee.ws.server.async.DefaultRequestQueue`.
JMS Response Queue Connection Factory Name	Name of the connection factory for the JMS response queue. The default JMS connection factory, `weblogic.jms.XAConnectionFactory`, provided with the base domain is used by default.
JMS Response Queue Name	Name of the request queue. The following queue is used by default: oracle.j2ee.ws.server.async.DefaultResponseQueue.
JMS System User	The user that is authorized to use the JMS queues. By default, this property is set to `OracleSystemUser`. **Note:** For most users, the `OracleSystemUser` is sufficient. However, if you need to change this user to another user in your security realm, you can do so using the instructions provided in "Changing the JMS System User for Asynchronous Web Services Using Fusion Middleware Control".

6. Click **Apply**.

4.1.4.3 Changing the JMS System User for Asynchronous Web Services Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

By default, the JMS System User is set as `OracleSystemUser`. For most users, this default value is sufficient.

To configure a custom user in your security realm using Fusion Middleware Control, use one of the following methods:

- Attach the `oracle/async_web_service_policy` configuration policy to the web service and override the `jms.access.user` configuration property, as described in

"Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the asynchronous policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Attach the `oracle/no_async_web_service_policy` to disable an asynchronous policy configured at a higher scope.

For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Use the Configuration tab, as described below.

To change the JMS System User using the Configuration tab:

1. Access the **Configuration** tab on the Web Service Endpoint page for the asynchronous web service as described in "Configuring Asynchronous Web Services Using Fusion Middleware Control".
2. Enter the name of the custom user in the **JMS System User** field and click **Apply**.

> **Note:**
>
> The custom user must exist in the security realm and have the permissions required to access the JMS resources.

3. Access the WebLogic Server Administration Console. To do so from Fusion Middleware Control, select the domain in the navigation pane. From the **WebLogic Domain** menu, select **WebLogic Server Administration Console**.
4. Log into the WebLogic Server Administration Console using a valid username and password with the required administrative privileges.
5. Click **Deployments** in the Domain Structure pane and navigate to the corresponding *`service`*`_AsynchRequestProcessorMDB` or *`service`*`_AsynchResponseProcessorMDB` MDBs. In these MDB names, *`service`* is the name of the asynchronous service for which you are changing the user name.
6. In the Change Center, select **Lock & Edit**.
7. Select the MDB name for the request or response MDB. (You will need to update the user name for both the request and response MDBs.) In the Settings page, select the **Configuration** tab.
8. In the Enterprise Bean Configuration section of the page, enter the custom user name in the **Run As Principal Name** field and click **Save**.

   Note that the user name you enter in this field must match the user name you entered for the JMS System User in Fusion Middleware Control.

   The configuration changes need to be saved in a new deployment plan.

9. Use the Save Deployment Plan Assistant to save the new deployment plan.
10. Repeat steps 7 and 8 for the second MDB. The changes are automatically saved to the new deployment plan.
11. In the Change Center, click **Activate Changes**.
12. Redeploy the application. For more information, see Deploying Web Service Applications .

4.1.4.4 Configuring Reliable Messaging Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

To enable web services reliable messaging using Fusion Middleware Control, attach the `oracle/reliable_messaging_policy` policy to the web service, as described in "Attaching Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the reliable messaging policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Attach the `oracle/no_reliable_messaging_policy` to disable a reliable messaging configured at a higher scope.

For more information about the reliable messaging policies, see "Reliable Messaging Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.4.5 Configuring Atomic Transactions Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section:
>
> - Apply to Oracle Infrastructure web services only.
> - Do not apply to RESTful SOA component services.

Web services support the WS-Coordination and WS-AtomicTransaction (WS-AT) specifications. Therefore, you can configure web services atomic transactions to enable interoperability between Oracle WebLogic Server and other vendor's transaction processing systems, such as WebSphere, Microsoft .NET, and so on.

To configure web services atomic transactions policy using Fusion Middleware Control, use one of the following methods:

- Use the Configuration or Properties tab, as described below.

  > **Note:**
  >
  > Configuration using the Properties tab is the only option available for SOA composites; the configuration policy, described below, is not available for SOA composites.

- Attach the `oracle/atomic_transaction_policy` policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the atomic transactions policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Attach the `oracle/no_atomic_transaction_policy` to disable an atomic transaction policy configured at a higher scope.

  For more information about the atomic transaction policies, see "Atomic Transaction Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To configure atomic transactions for a web service using the Configuration or Properties tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".
2. Click the **Configuration** tab. For SOA composites (SOAP only), click the Properties tab.
3. In the **Atomic Transaction Version** field, select the version of the web service atomic transaction coordination context that is supported. The value specified must be consistent across the entire transaction. Valid values are:
   - **WSAT10**
   - **WSAT11**
   - **WSAT12**
   - **Default**

   If you select **Default**, all three versions are accepted.

> **Note:**
>
> This property works with SOA web services that have synchronous-only operations and with web services that have both synchronous and asynchronous operations. It does not work with SOA web services with asynchronous-only operations.

4. In the **Atomic Transaction Flow Option** field, select whether the transaction coordination context is to be passed with the transaction flow into the web service.

   Valid values include:

   - **Never**—Do not export transaction coordination context. This is the default.
   - **Supports**—Export transaction coordination context if transaction is available.
   - **Mandatory**—Export transaction coordination context. An exception is thrown if there is no active transaction.

5. Click **Apply**.

4.1.4.6 Configuring MTOM Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to Oracle Infrastructure web services only.

To enable MTOM using Fusion Middleware Control, attach the `oracle/wsmtom_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the MTOM policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Attach the `oracle/no_mtom_policy` to disable an MTOM policy configured at a higher scope.

For more information about the MTOM policies, see "MTOM Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.4.7 Configuring Fast Infoset Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

To enable Fast Infoset using Fusion Middleware Control, attach the `oracle/fast_infoset_service_policy` policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the Fast Infoset policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Attach the `oracle/no_fast_infoset_service_policy` to disable a Fast Infoset policy configured at a higher scope.

For more information about the Fast Infoset policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.4.8 Configuring Persistence Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section applies to non-SOA Oracle Infrastructure web services only.

To enable persistence using Fusion Middleware Control, attach the `oracle/persistence_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the persistence policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Attach the `oracle/no_persistence_policy` to disable a persistence policy configured at a higher scope.

For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.4.9 Configuring SOAP Over JMS Transport Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

Typically, web services and clients communicate using SOAP over HTTP/S as the connection protocol. You can, however, configure a web service so that client applications use JMS as the transport.

Using SOAP over JMS transport, web services and clients communicate using JMS destinations instead of HTTP connections, offering the following benefits:

- Reliability
- Scalability
- Quality of service

As with web service reliable messaging, if WebLogic Server goes down while the method invocation is still in the queue, it will be handled as soon as WebLogic Server is restarted. When a client invokes a web service, the client does not wait for a response, and the execution of the client can continue. Using SOAP over JMS transport does require slightly more overhead and programming complexity than HTTP/S.

You can enable and configure SOAP over JMS transport at design time using the `@JMSTransportService` annotation, as described in "Security and Policy Annotation Reference" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To configure SOAP over JMS transport policy using Fusion Middleware Control, use one of the following methods:

- Attach the `oracle/jms_transport_service_policy` policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the JMS transport policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  - Attach the `oracle/no_jms_transport_service_policy` to disable a SOAP over JMS transport policy configured at a higher scope.

  For more information about the SOAP over JMS transport policies, see "SOAP Over JMS Transport Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Use the Configuration tab, as described below.

To configure SOAP over JMS transport using the Configuration tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".

   When SOAP over JMS transport is enabled, the SOAP over JMS transport configuration properties are displayed in the summary area. The configuration properties are read-only.

2. Click the **Configuration** tab.
3. In the SOAP Over JMS Enable HTTP WSDL Access field, select **True** from the menu to enable WSDL access or select **False** to disable WSDL access.
4. Click **Apply**.

4.1.4.10 Enabling or Disabling Web Services Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

To enable or disable web services using Fusion Middleware Control, use one of the following methods:

- Attach the `oracle/request_processing_service_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the web service request processing policy, perform one of the following steps:

  - Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
  - Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
  - Attach the `oracle/no_request_processing_policy` to disable a web service request processing policy configured at a higher scope.

  For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Use the Configuration tab, as described below.

To enable or disable a web service endpoint using the Configuration tab:

1. Navigate to the Web Services Application Summary page.
2. In the Web Service Details section of the page, expand the web service to display the web service endpoints if they are not already displayed.
3. Click the name of the endpoint to navigate to the Web Service Endpoint page.
4. From the Web Service Endpoint page, click the **Configuration** tab.
5. In the Endpoint Enabled field, select **Enabled** or **Disabled** from the menu to enable or disable the web service, respectively.
6. Click **Apply.**

4.1.4.11 Enabling or Disabling Public Access to the Web Service WSDL Document Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section:
>
> - Apply to Oracle Infrastructure web services only.
> - Do not apply to RESTful SOA component services.

To enable public access to the web service WSDL document using Fusion Middleware Control, use one of the following methods:

- Use the Configuration or Properties tab, as described below.

  > **Note:**
  >
  > Configuration using the Properties tab is the only option available for SOA composites; the configuration policy, described below, is not available for SOA composites.

- Attach the `oracle/wsdl_request_processing_service_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the WSDL access policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Attach the `oracle/no_wsdl_request_processing_service_policy` to disable a WSDL access policy configured at a higher scope.

For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To enable or disable the display of the web service WSDL document using the Configuration or Properties tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".
2. Click the **Configuration** tab. For SOA composites (SOAP only), click the Properties tab.
3. From the WSDL Enabled field, select **True** from the menu to enable the display of the WSDL or **False** to disable the display of the WSDL.
4. Click **Apply.**

4.1.4.12 Enabling or Disabling SOAP Processing Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

To enable the processing of SOAP requests on the web service endpoint using Fusion Middleware Control, attach the `oracle/soap_request_processing_service_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the SOAP processing policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Attach the `oracle/no_request_processing_service_policy` to disable a SOAP processing policy configured at a higher scope.

For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.4.13 Enabling or Disabling Non-SOAP XML Message Processing Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section:
>
> - Apply to Oracle Infrastructure web services only.
> - Do not apply to RESTful SOA component services.

To enable an endpoint to receive non-SOAP XML messages that are processed by a user-defined `javax.xml.ws.Provider<T>.invoke` method using Fusion Middleware Control, use one of the following methods:

- Use the Configuration or Properties tab, as described below.

  > **Note:**
  >
  > Configuration using the Properties tab is the only option available for SOA composites; the configuration policy, described below, is not available for SOA composites.

- Attach the `oracle/pox_http_binding_service_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the non-SOAP XML message processing policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Attach the `oracle/no_pox_http_binding_service_policy` to disable a a non-SOAP XML message processing policy configured at a higher scope.

  For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To enable non-SOAP XML message processing using the Configuration or Properties tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".

2. Click the **Configuration** tab. For SOA composites (SOAP only), click the Properties tab.
3. In the RESTful Enabled field, select **True** from the menu to enable the feature, or select **False** to disable the feature.
4. Click **Apply**.

4.1.4.14 Setting the Log Level for Diagnostic Logs Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section:
>
> - Apply to Oracle Infrastructure web services only.
> - Do not apply to RESTful SOA component services.

To set the log level for diagnostic logs using Fusion Middleware Control, use one of the following methods:

- Use the Configuration or Properties tab, as described below.

  > **Note:**
  >
  > Configuration using the Properties tab is the only option available for SOA composites; the configuration policy, described below, is not available for SOA composites.

- Attach the `oracle/ws_logging_level_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the logging level policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Attach the `oracle/no_ws_logging_level_service_policy` to disable a logging level policy configured at a higher scope.

  For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To set the logging level for diagnostic logs using the Configuration or Properties tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".
2. Click the **Configuration** tab. For SOA composites (SOAP only), click the Properties tab.
3. In the Logging Level field, select the logging level.
4. Click **Apply**.

4.1.4.15 Introduction to Enabling or Disabling the Web Services Test Client Using Fusion Middleware Control

You can enable or disable the Web Service Test Client at the domain or web service endpoint level using Fusion Middleware Control, as described in the following sections:

- "Enabling or Disabling the Web Services Test Client at the Domain Level Using Fusion Middleware Control"
- "Enabling or Disabling the Web Service Test Client at the Web Service Endpoint Level Using Fusion Middleware Control"

For more information about the Web Services Test Client, see "Using the Web Services Test Client".

> **Note:**
>
> The procedures described in this section do not impact the availability of the **Web Services Test** link on the Web Service Endpoint page, which enables you to access the Fusion Middleware Control Test Web Service page. For more information, see "Test Web Service Page in Fusion Middleware Control".

4.1.4.15.1 Enabling or Disabling the Web Services Test Client at the Domain Level Using Fusion Middleware Control

To enable or disable the Web Services Test Client at the domain level using Fusion Middleware Control:

1. Select **WebLogic Domain > Administration > General Settings**.
2. Click **Advanced** to display the advanced settings.
3. Toggle the **Enable Web Service Test Page** flag to enable or disable the Web Services Test Client at the domain level.
4. Click **Save**.
5. Restart the WebLogic domain.

4.1.4.15.1.1 Enabling or Disabling the Web Service Test Client at the Web Service Endpoint Level Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section:
>
> - Apply to Oracle Infrastructure web services only.
> - Do not apply to RESTful SOA component services.

To enable the Web Services Test Client using Fusion Middleware Control, use one of the following methods:

- Use the Configuration or Properties tab, as described below.

  > **Note:**
  >
  > Configuration using the Properties tab is the only option available for SOA composites (SOAP only); the configuration policy, described below, is not available for SOA composites.

- Attach the `oracle/test_page_processing_service_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the Web Services Test Client policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Attach the `oracle/no_ws_logging_level_service_policy` to disable a Web Services Test Client policy configured at a higher scope.

  For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To enable the Web Service Test Client using the Configuration or Properties tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".
2. Click the **Configuration** tab. For SOA composites (SOAP only), click the Properties tab.
3. In the Endpoint Test Enabled field, select **True** from the menu to enable the test endpoint or **False** to disable the test endpoint.

4. Click **Apply**.

4.1.4.16 Enabling or Disabling the Exchange of Metadata Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section:
>
> - Apply to Oracle Infrastructure web services only.
> - Do not apply to RESTful SOA component services.

To enable the exchange of metadata using Fusion Middleware Control, use one of the following methods:

- Use the Configuration or Properties tab, as described below.

  > **Note:**
  >
  > Configuration using the Properties tab is the only option available for SOA composites (SOAP only); the configuration policy, described below, is not available for SOA composites.

- Attach the `oracle/mex_request_processing_service_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the metadata exchange policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Attach the `oracle/no_mex_request_processing_service_policy` to disable a metadata exchange policy configured at a higher scope.

  For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To enable or disable the exchange of metadata using the Configuration or Properties tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".

2. Click the **Configuration** tab. For SOA composites (SOAP only), click the Properties tab.
3. In the Metadata Exchange Enabled field, select **True** from the menu to enable the exchange of metadata or **False** to disable the exchange of metadata.
4. Click **Apply**.

4.1.4.17 Configuring MTOM-encoded Fault Messages Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

To enable MTOM-encoded fault messages when MTOM is enabled using Fusion Middleware Control, attach the `oracle/mtom_encode_fault_service_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the MTOM-encoded fault processing policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Attach the `oracle/no_mtom_encode_fault_service_policy` to disable an MTOM-encoded fault processing policy configured at a higher scope.

For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.4.18 Validating the Request Message Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section:
>
> - Apply to Oracle Infrastructure web services only.
> - Do not apply to RESTful SOA component services.

To validate the request message against the schema using Fusion Middleware Control, use one of the following methods:

- Use the Configuration or Properties tab, as described below.

  > **Note:**
  >
  > Configuration using the Properties tab is the only option available for SOA composites (SOAP only); the configuration policy, described below, is not available for SOA composites.

- Attach the `oracle/schema_validation_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the schema validation policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Attach the `oracle/no_schema_validation_policy` to disable a schema validation policy configured at a higher scope.

  For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To enable or disable schema validation using the Configuration or Properties tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".
2. Click the **Configuration** tab. For SOA composites (SOAP only), click the Properties tab.
3. In the Schema validation field, select **True** from the menu to enable schema validation or **False** to disable schema validation.
4. Click **Apply**.

4.1.4.19 Setting the Size of the Request Message Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section:
>
> - Apply to Oracle Infrastructure web services only.
> - Do not apply to RESTful SOA component services.

To set the maximum size of the request message using Fusion Middleware Control, use one of the following methods:

- Use the Configuration or Properties tab, as described below.

> **Note:**
>
> Configuration using the Properties tab is the only option available for SOA composites (SOAP only); the configuration policy, described below, is not available for SOA composites.

- Attach the `oracle/max_request_size_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  To disable the maximum request size policy, perform one of the following steps:

  – Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  – Attach the `oracle/no_max_request_size_policy` to disable a maximum request size policy configured at a higher scope.

  For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To set the size of the request message using the Configuration or Properties tab:

1. Navigate to the Web Service Endpoint page, or the Service Home page (for SOA composites), as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".
2. Click the **Configuration** tab. For SOA composites (SOAP only), click the Properties tab.
3. Set the Maximum Request Size and the Unit of Maximum Request Size.

   -1 sets no limit to the size of the message. Or, you can set a maximum limit to the message by entering a number in the text box and selecting the unit of measurement.

> **Note:**
>
> If you set the Maximum Request Size to -1, indicating that there is no maximum request size, then the Unit of Maximum Request Size setting is irrelevant and defaults to bytes.

4. Click **Apply**.

4.1.4.20 Enabling or Disabling Binary Content Caching Using Fusion Middleware Control

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services only.

To enable binary content caching using Fusion Middleware Control, attach the `oracle/cache_binary_content_policy` configuration policy to the web service, as described in "Attaching Policies to Web Services and Clients Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the binary caching content policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Attach the `oracle/no_cache_binary_content_policy` to disable a binary caching content policy configured at a higher scope.

For more information about the configuration policies, see "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

4.1.5 Overview of Configuring Web Service Clients Using Fusion Middleware Control

You can use the Fusion Middleware Control to configure SOA references, ADF DC and asynchronous web service callback clients.

> **Note:**
>
> The procedures described in this section apply to Oracle Infrastructure web service clients only.

For more details, refer to the following procedures.

- Configuration Properties for Web Service Clients
- Configuring SOA References
- Configuring Connection-based Web Service Clients
- Configuring Asynchronous Web Service Callback Clients

4.1.5.1 Configuration Properties for Web Service Clients

For the web service clients in your application, including SOA references, ADF data control, and asynchronous web service Callback clients, you can set the configuration properties defined in Table 4-3.

Table 4-3 Configuration Properties for Web Service Clients

Configuration Property	Property Name	Description
General		
UDDI ServiceKey	oracle.soa.uddi.serviceKey	Specifies the service key of the Oracle Service Registry (OSR) if UDDI is used for run-time resolution of the endpoint. For more information, see "Changing the Endpoint Reference and Service Key for Oracle Service Registry Integration" in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*. **Note**: This property is available for SOAP SOA reference clients only.
Endpoint Address	`javax.xml.ws.service.endpoint.address`	Endpoint URL to which the client will send the request. **Note**: This property is not available for asynchronous web service Callback clients.
WS Addressing Reply To	`oracle.webservices.wsaddressing.replyTo`	Specifies a callback URL for the ADF Web Services Data Control/Web Services Connection client. The value is used in the WS-addressing `replyTo` header in the outgoing message. **Note**: This property is not available for RESTful SOA reference clients.
Maintain Session	`javax.xml.ws.session.maintain`	Flag that specifies whether the session should be maintained. **Note**: This property is not available for asynchronous web service Callback clients or RESTful SOA reference clients.

Table 4-3 (Cont.) Configuration Properties for Web Service Clients

Configuration Property	Property Name	Description
Atomic Transaction Version	wsat.Version	Specifies the version of the SOA web service atomic transaction coordination context used for outbound messages only. The value specified must be consistent across the entire transaction. Valid values are **WSAT10**, **WSAT11**, **WSAT12**, and **Default**. Note that if the flow option is set to **WSDL Driven**, you cannot specify a version. The version advertised in the WSDL is used. If the flow option is set to **Supports** or **Mandatory** and you specify the **Default** option, then WSAT10 is used. **Note:** In WLST, the valid values must be specified as "WSAT10", "WSAT11", "WSAT12", and "DEFAULT". Use of an invalid value results in an error message. **Note**: This property is applicable for SOAP SOA reference clients only.
Atomic Transaction Flow Option	wsat.flowOption	Specifies whether the transaction coordination context is passed with the transaction flow. Valid values on the SOA reference client are: • **Never** (default) – Do not export transaction coordination context. • **Supports** – Export transaction coordination context if transaction is available. • **Mandatory** – Export transaction coordination context. An exception is thrown if there is no active transaction. • **WSDL Driven** – Use the value set in the WSDL. **Note:** In WLST, the valid values must be specified as "NEVER", "SUPPORTS", "MANDATORY", and "WSDLDriven". Use of an invalid value results in an error message. **Note**: This property is applicable for SOAP SOA reference clients only.
HTTP Chunking		
Stop Chunking	`oracle.webservices.donotChunk`	Flag that specifies whether chunking is enabled for client requests.
Chunking Size (bytes)	`oracle.webservices.chunkSize`	Size of the request chunk in bytes.

Table 4-3 (Cont.) Configuration Properties for Web Service Clients

Configuration Property	Property Name	Description
HTTP Timeout		
HTTP Read Timeout (ms)	`oracle.webservices.httpReadTimeout`	Length of the request read timeout in milliseconds.
HTTP Connection Timeout (ms)	`oracle.webservices.httpConnTimeout`	Length of the request connection timeout in milliseconds.
HTTP Basic Authentication		
HTTP User Name	`(javax.xml.ws.security.auth.username)` `oracle.webservices.auth.username`	Authenticated HTTP user name. **Note**: This property is not available for RESTful SOA reference clients.
HTTP User Password	`(javax.xml.ws.security.auth.password)` `oracle.webservices.auth.password`	Authenticated HTTP user password. **Note**: This property is not available for RESTful SOA reference clients.
Preemptive	`oracle.webservices.preemptiveBasicAuth`	Flag that specifies whether security will be sent with the request without being challenged. **Note**: This property is not available for RESTful SOA reference clients.
HTTP Proxy		
Proxy Host	`oracle.webservices.proxyHost`	URL of proxy to which client will send the request. **Note**: This property is not available for RESTful SOA reference clients.
Proxy Port	`oracle.webservices.proxyPort`	Port number of the proxy. **Note**: This property is not available for RESTful SOA reference clients.
Proxy User Name	`oracle.webservices.proxyUsername`	Valid user name to access the proxy. **Note**: This property is not available for RESTful SOA reference clients.
Proxy User Password	`oracle.webservices.proxyPassword`	Valid password to access the proxy. **Note**: This property is not available for RESTful SOA reference clients.
Proxy Realm	`oracle.webservices.proxyAuthRealm`	Realm used by the proxy. **Note**: This property is not available for RESTful SOA reference clients.
Proxy Authentication Type	`oracle.webservices.proxyAuthType`	Authentication type used by the proxy. **Note**: This property is not available for RESTful SOA reference clients.

4.1.5.2 Configuring SOA References

The following procedure describes how to configure a SOA reference.

1. View the SOA reference, as described in "Viewing SOA References".

2. Click the **Properties** tab.
3. Set the property values as required. Refer to Table 4-3.
4. Click **Apply**.

4.1.5.3 Configuring Connection-based Web Service Clients

The following procedure describes how to configure a connection-based web service client such as an ADF DC web service client or ADF JAX-WS Indirection Proxy.

1. View the connection-based web service client as described in "Viewing Connection-Based Web Service Clients".
2. Click the **Configuration** tab.
3. Set the configuration values as required. Refer to Table 4-3.
4. Click **Apply**.

4.1.5.4 Configuring Asynchronous Web Service Callback Clients

The following procedure describes how to configure an asynchronous web service Callback client. Callback clients are used only by asynchronous web services to return the response to the caller. For more information, see "Developing Asynchronous Web Services" in *Developing Oracle Infrastructure Web Services*.

To configure an asynchronous web service callback client:

1. Navigate to the endpoint for the asynchronous web service, as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".
2. Click **Callback Client** in the upper right portion of the endpoint page.
3. Click the **Configuration** tab.
4. Set the configuration values as required. Refer to Table 4-3.
5. Click **Apply**.

4.1.6 Overview of Managing the WSDL Using Fusion Middleware Control

In some cases, you might not want the web service WSDL to be accessible to the public. You can enable or disable public access to the WSDL from the Web Service Endpoint page.

> **Note:**
>
> In some cases, a web service client needs to access a WSDL during invocation. If public access to the WSDL is disabled, the client will need to have a local copy of the WSDL.

- Viewing the Web Service WSDL Document
- Enabling or Disabling Public Access to the Web Service WSDL Document

4.1.6.1 Viewing the Web Service WSDL Document

To display the WSDL document for a web service:

1. Navigate to the Web Services Application Summary page.
2. In the Web Service Details section of the page, expand the web service to display the web service endpoints if they are not already displayed.
3. Click the name of the endpoint to navigate to the Web Service Endpoint page.
4. In the WSDL Document field, click the endpoint name to display the WSDL for the web service.

4.1.6.2 Enabling or Disabling Public Access to the Web Service WSDL Document

To enable or disable the display of the web service WSDL document:

- Using Fusion Middleware Control, see "Enabling or Disabling Public Access to the Web Service WSDL Document Using Fusion Middleware Control".
- Using WLST, see "Enabling or Disabling Public Access to the Web Service WSDL Document Using WLST".

4.2 Overview of Web Services Administration Using WLST

You can use WLST to perform common web services administration tasks.

- Viewing the Web Services in a Domain Using WLST
- Viewing the Web Services in Your Application Using WLST
- Viewing the Details for a Web Service Endpoint Using WLST
- Viewing Web Service Clients Using WLST
- Overview of Configuring Web Services Using WLST
- Configuring Web Service Clients Using WLST

4.2.1 Viewing the Web Services in a Domain Using WLST

To view all the current web services in a domain:

1. Connect to the running instance of WebLogic Server for which you want to view the web services as described in "Accessing the Web Services Custom WLST Commands".
2. Use either the `listWebServices()` or the `listWSMPolicySubjects()` WLST command to display a list of the web services. If you do not specify a web service application or a SOA composite, the command lists all services in all applications and composites for every server instance in the domain.

   This is an example of the `listWebServices()` command:

   ```
   wls:/base_domain/serverConfig> listWebServices()

   /base_domain/AdminServer/jaxwsejb30ws :
   ```

```
        moduleName=jaxwsejb, moduleType=web,
serviceName=JaxwsWithHandlerChainBeanService
        moduleName=jaxwsejb, moduleType=web, serviceName=WsdlConcreteService
        moduleName=jaxwsejb, moduleType=web, serviceName=EchoEJBService
        moduleName=jaxwsejb, moduleType=web, serviceName=CalculatorService
        moduleName=jaxwsejb, moduleType=web, serviceName=DoclitWrapperWTJService

/base_domain/AdminServer/webservicesJwsSimple :
        moduleName=webservicesJwsSimple!SimpleImplService, moduleType=wls,
serviceName=examples.webservices.jws_basic.simple.SimpleImpl

/base_domain/AdminServer/CalWSBA :
        moduleName=CalWSBA#1!CalculatorService, moduleType=wls,
serviceName=CalculatorService

/base_domain/AdminServer/SimpleRestApp :
        moduleName=SimpleRestApp, moduleType=web,
serviceName=SimpleRestServiceService

/base_domain/AdminServer/SimpleJAXWS :
        moduleName=SimpleJAXWS#1!SimpleImplService, moduleType=wls,
serviceName=SimpleImplService

        moduleName=SimpleJAXWS#1!SimpleEjbService, moduleType=wls,
serviceName=SimpleEjbService
```

This is an example of the `listWSMPolicySubjects()` command:

```
wls:/base_domain/serverConfig> listWSMPolicySubjects()

Application: /weblogic/base_domain/SimpleRestApp

  Assembly: #SimpleRestApp

    Subject: WS-Service({http://
rest.jaxws.ws.j2ee.oracle/}SimpleRestServiceService#SimpleRestServicePort)

Application: /weblogic/base_domain/jaxwsejb30ws

  Assembly: #jaxwsejb

    Subject: WS-Service({http://host.example.com/
targetNamespace}EchoEJBService#EchoEJBServicePort)

    Subject: WS-Service({http://host.examplecom/jaxws/tests/
concrete}WsdlConcreteService#WsdlConcretePort)

    Subject: WS-Service({http://host.example.com/jaxws/
tests}CalculatorService#CalculatorPort)

    Subject: WS-Service({http://soapinterop.org/
DoclitWrapperWTJ}DoclitWrapperWTJService#DoclitWrapperWTJPort)

    Subject: WS-Service({http://
j2ee.tests.ejb.impl/}JaxwsWithHandlerChainBeanService#JaxwsWithHandlerChainBeanPo
rt)

Application: /weblogic/base_domain/em

  Assembly: #default
```

```
    Subject: WS-Client({http://host.example.com/jaxws/
tests}CalculatorService#CalculatorPort)
```

3. Set the `detail` argument of the `listWSMPolicySubjects` or `listWebServices` command to `true` to view the endpoint (port) and policy details for all applications and composites in the domain, the secure status of the endpoints, any configuration overrides and constraints, and if the endpoints have a valid configuration. Because you can specify the priority of a global or directly attached policy (using the `reference.priority` configuration override), the `effective` field indicates if directly attached policies are in effect for the endpoint.

> **Note:**
>
> To simplify endpoint management, all directly attached policies are shown in the output regardless of whether they are in effect for the endpoint. In contrast, only globally attached policies that are in effect for the endpoint are displayed.

An endpoint is considered secure if the policies attached to it (either directly or externally) enforce authentication, authorization, or message protection behaviors.

> **Note:**
>
> The `listWebServices` command output does not include details on SOA components, including policy attachments.

```
wls:/base_domain/serverConfig> listWebServices(detail='true')
/weblogic/base_domain/jaxwsejb30ws :
        moduleName=jaxwsejb, moduleType=web,
serviceName=JaxwsWithHandlerChainBeanService
                JaxwsWithHandlerChainBeanPort
http://host.example.com:17001/jaxwsejb/JaxwsWithHandlerChainIntf
                URI="oracle/mex_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/mtom_encode_fault_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/max_request_size_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                        Property name="max.request.size", value="-1"
                URI="oracle/request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
```

```
                URI="oracle/soap_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/ws_logging_level_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                        Property name="logging.level", value=""
                URI="oracle/test_page_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/wsdl_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/wss_saml_or_username_token_over_ssl_service_
policy", category=security, policy-status=enabled; source=global policy set
"test", scope="DOMAIN('*')"; reference-status=enabled; effective=true

        The policy subject is secure in this context.

        moduleName=jaxwsejb, moduleType=web, serviceName=WsdlConcreteService
                WsdlConcretePort
http://host.example.com:17001/jaxwsejb/WsdlAbstract
                URI="oracle/mex_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/mtom_encode_fault_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/max_request_size_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                        Property name="max.request.size", value="-1"
                URI="oracle/request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/soap_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/ws_logging_level_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
```

```
                        Property name="logging.level", value=""
                URI="oracle/test_page_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/wsdl_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true
                        Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                URI="oracle/wss_saml_or_username_token_over_ssl_service_
policy", category=security, policy-status=enabled; source=global policy set
"test", scope="DOMAIN('*')"; reference-status=enabled; effective=true

        The policy subject is secure in this context.
```

This is an example for the `listWSMPolicySubjects` command. It lists the detail output for the `WsdlConcretePort` subject.

```
wls:/base_domain/serverConfig> listWSMPolicySubjects ('jaxwsejb30ws',
'#jaxwsejb', None, detail='true')
Application: /WLS/rc6_domain/jaxwsejb30ws

  Assembly: #jaxwsejb

    Subject: WS-Service({http://www.oracle.com/jaxws/tests/
concrete}WsdlConcreteService#WsdlConcretePort)

        URI="oracle/mex_request_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/mtom_encode_fault_service_policy", category=wsconfig, policy-
status=enabled; source=local policy set; reference-status=enabled; effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/max_request_size_policy", category=wsconfig, policy-
status=enabled; source=local policy set; reference-status=enabled; effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                Property name="max.request.size", value="-1"
        URI="oracle/request_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/soap_request_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/ws_logging_level_policy", category=wsconfig, policy-
status=enabled; source=local policy set; reference-status=enabled; effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                Property name="logging.level", value=""
        URI="oracle/test_page_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
```

```
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/wsdl_request_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"

        The policy subject is not secure in this context.
    Subject: WS-Service({http://
oracle.j2ee.tests.ejb.impl/}JaxwsWithHandlerChainBeanService#JaxwsWithHandlerChai
nBeanPort)

        URI="oracle/mex_request_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/mtom_encode_fault_service_policy", category=wsconfig, policy-
status=enabled; source=local policy set; reference-status=enabled; effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/max_request_size_policy", category=wsconfig, policy-
status=enabled; source=local policy set; reference-status=enabled; effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                Property name="max.request.size", value="-1"
        URI="oracle/request_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/soap_request_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/ws_logging_level_policy", category=wsconfig, policy-
status=enabled; source=local policy set; reference-status=enabled; effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
                Property name="logging.level", value=""
        URI="oracle/test_page_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"
        URI="oracle/wsdl_request_processing_service_policy", category=wsconfig,
policy-status=enabled; source=local policy set; reference-status=enabled;
effective=true
                Property name="local.policy.reference.source",
value="IMPLIED_FEATURE"

        The policy subject is not secure in this context.
...
```

For more information about the `listWSMPolicySubjects` and `listWebServices` commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.2 Viewing the Web Services in Your Application Using WLST

To view the web services in your application:

1. Connect to the running instance of WebLogic Server to which the application is deployed as described in "Accessing the Web Services Custom WLST Commands".
2. Use the `listWebServices` WLST command to display a list of the web services in your application. You must specify the complete application path name to identify the application and the server instance to which it is deployed.

   ```
   listWebServices (application,composite,[detail]
   ```

   For example:

   ```
   wls:/wls-domain/serverConfig>listWebServices("wls-domain/AdminServer/
   jaxwsejb30ws")
   /wls-domain/AdminServer/jaxwsejb30ws:
     moduleName=jaxwsejb,moduleType=web,serviceName={http://
   namespace/}JaxwsWithHandlerChainBeanService
    moduleName=jaxwsejb, moduleType=web, serviceName={http://
   namespace/}WsdlConcreteService
    moduleName=jaxwsejb, moduleType=web, serviceName={http://
   namespace/}EchoEJBService
    moduleName=jaxwsejb, moduleType=web, serviceName={http://
   namespace/}CalculatorService
    moduleName=jaxwsejb, moduleType=web, serviceName={http://
   namespace/}DoclitWrapperWTJService
   ```

For details about the `listWebServices` command, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.3 Viewing the Details for a Web Service Endpoint Using WLST

To view the details for a web service endpoint (port):

1. Connect to the running instance of WebLogic Server to which the application is deployed as described in "Accessing the Web Services Custom WLST Commands".
2. Use the `listWebServices` WLST command to display a list of the web services in your application as described in "Viewing the Web Services in Your Application Using WLST".
3. Use the `listWebServicePorts` command to display the endpoint name and endpoint URL for a web service.

   ```
   listWebServicePorts(application,moduleOrCompName,moduleType,serviceName)
   ```

   For example, to display the endpoint for the `WsdlConcreteService`:

   ```
   wls:/wls-domain/serverConfig> listWebServicePorts
   ('jaxwsejb30ws','jaxwsejb','web','WsdlConcreteService')

   WsdlConcretePort   http://host.example.com:7001/jaxwsejb/WsdlAbstract
   ```

4. Use the `listWebServicePolicies` command to view the policies that are attached to a web service endpoint.

```
listWebServicePolicies(application,moduleOrCompName,moduleType,serviceName,subjec
tName)
```

For example, to view the policies attached to the `WsdlConcretePort` endpoint and any policy override settings:

```
wls:/wls_domain/serverConfig> listWebServicePolicies ("jaxwsejb30ws",
"jaxwsejb","web","WsdlConcreteService","WsdlConcretePort")

WsdlConcretePort :
                URI="oracle/mex_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set; reference-
status=enabled; effective=true
                URI="oracle/mtom_encode_fault_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set; reference-
status=enabled; effective=true
                URI="oracle/max_request_size_policy", category=wsconfig, policy-
status=enabled; source=local policy set; reference-status=enabled; effective=true
                        Property name="max.request.size", value="-1"
                URI="oracle/request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set; reference-
status=enabled; effective=true
                URI="oracle/soap_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set; reference-
status=enabled; effective=true
                URI="oracle/ws_logging_level_policy", category=wsconfig, policy-
status=enabled; source=local policy set; reference-status=enabled; effective=true
                        Property name="logging.level", value=""
                URI="oracle/test_page_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set; reference-
status=enabled; effective=true
                URI="oracle/wsdl_request_processing_service_policy",
category=wsconfig, policy-status=enabled; source=local policy set; reference-
status=enabled; effective=true
                URI="oracle/wss_saml_or_username_token_over_ssl_service_policy",
category=security, policy-status=enabled; source=global policy set "test_PS",
scope="DOMAIN('*')"; reference-status=enabled; effective=true

        The policy subject is secure in this context.
```

For more information about these WLST commands and their arguments, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.4 Viewing Web Service Clients Using WLST

Use the following procedure to view the web service clients using WLST commands:

1. Connect to the running instance of WebLogic Server to which the application is deployed as described in "Accessing the Web Services Custom WLST Commands".
2. Use the `listWebServiceClients` WLST command to display a list of the web service clients.

   ```
   listWebServiceClients(application,composite,[detail])
   ```

   This command enables you to list the clients for an application, a SOA composite, or a domain. To list the client information for an application or SOA composite, specify the appropriate argument. If you do not specify an application or SOA

composite, the command outputs information, including the module name, module type, and SOA reference name for all the web service clients in all applications and composites in every server instance in the domain. To view details about each client, including the endpoint and policies, set the `detail` argument to `true`.

For example:

```
wls:/soainfra/serverConfig> listWebServiceClients(detail='true')



/soainfra/soa_server1/soa-infra :

        compositeName=default/SampleSOAFirstPrj[1.0], moduleType=soa,
serviceRefName=ReferenceToSecondSOA

                BPELProcess1_pt   serviceWSDLURI=

                  http://localhost:8001/soa-infra/services/default/

                  SampleSOASecondPrj/BPELProcess1.wsdl

                oracle.webservices.contentTransferEncoding=base64

                oracle.webservices.charsetEncoding=UTF-8

                oracle.webservices.operationStyleProperty=document

                oracle.webservices.soapVersion=soap1.1

                oracle.webservices.chunkSize=4096

                oracle.webservices.preemptiveBasicAuth=false

                oracle.webservices.session.maintain=false

                oracle.webservices.encodingStyleProperty=

                  http://schemas.xmlsoap.org/soap/encoding/

                oracle.webservices.donotChunk=true

                No attached policies found; endpoint is not secure.



/soainfra/AdminServer/ADFDCDecoupling_Project1_ADFDCDecoupling-1 :
        moduleName=testadfbc, moduleType=wsconn,
serviceRefName=AppModuleService
                AppModuleServiceSoapHttpPort
serviceWSDLURI=http://adc1140275.example.com:7001/ADFBCDecoupling-
ADFBCDecoupling-context-root/AppModuleService?wsdl
                URI="oracle/wss10_saml_token_with_message_protection_client_
policy", category=security, policy-status=enabled; source=local policy set;
reference-status=enabled; effective=true

        The policy subject is secure in this context.
```

Note that the output displays SOA references (using the `serviceRefName` argument) for the SOA composites `default/SampleSOAFirstPrj[1.0]`. To list the SOA references for a SOA composite, specify the composite name in the

command, for example `listWebServiceClients(None,'default/SampleSOAFirstPrj[1.0]')`.

ADF clients are specified by the `moduleType=wsconn` argument in the output.

For more information about the WLST commands and their arguments, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5 Overview of Configuring Web Services Using WLST

> **Note:**
>
> The procedures described in this section apply to non-SOA Oracle Infrastructure web services and providers only.
>
> Oracle Infrastructure web service providers implement the `java.xml.ws.Provider` interface. On the Web Service Endpoint page, they display the Implementation Class and provide a subset of configuration properties.

You configure the web service endpoint using WLST by attaching one of the configuration policies defined in "Configuration Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

The following sections describe how to perform common web services configuration tasks using WLST.

- Configuring Addressing Using WLST
- Configuring Asynchronous Web Services Using WLST
- Configuring the JMS System User for Asynchronous Web Services Using WLST
- Configuring Reliable Messaging Using WLST
- Configuring Atomic Transactions Using WLST
- Configuring MTOM Using WLST
- Configuring Fast Infoset Using WLST
- Configuring SOAP Over JMS Transport Using WLST
- Configuring Persistence Using WLST
- Enabling or Disabling Web Services Using WLST
- Enabling or Disabling Public Access to the Web Service WSDL Document Using WLST
- Enabling or Disabling the Processing of SOAP Requests Using WLST
- Enabling or Disabling Non-SOAP XML Message Processing Using WLST
- Setting the Log Level for Diagnostic Logs Using WLST
- Overview of Enabling or Disabling the Web Services Test Client Using WLST
- Enabling or Disabling MTOM-encoded SOAP Fault Messages Using WLST

- Validating the Request Message Using WLST
- Setting the Maximum Size of the Request Message Using WLST
- Configuring Binary Caching of Content
- Configuring Virtual User Using WLST

4.2.5.1 Configuring Addressing Using WLST

To configure web services addressing using WLST:

1. Attach the `oracle/wsaddr_policy` policy using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/wsaddr_policy")

   Policy reference "oracle/wsaddr_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/wsaddr_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   wsaddr_policy','reference.priority','10')

   The configuration override property "reference.priority" having value "10" has
   been added to the reference to policy with URI "oracle/wsaddr_policy".
   ```

   For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the addressing policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable an addressing policy configured at a higher scope, attach the `oracle/no_addressing_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_addressing_policy")

Policy reference "oracle/no_addressing_policy" added.
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.2 Configuring Asynchronous Web Services Using WLST

To configure asynchronous web services:

1. Attach the `oracle/async_web_service_policy` policy using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   async_web_service_policy")

   Policy reference "oracle/async_web_service_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/async_web_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   async_web_service_policy','jms.queue','myDefaultRequestQueue')

   The configuration override property "jms.queue" having value
   "myDefaultRequestQueue" has been added to the reference to policy with URI
   "oracle/async_web_service_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the asynchronous policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable an asynchronous policy configured at a higher scope, attach the `oracle/no_async_web_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_async_web_service_policy")

Policy reference "oracle/no_async_web_service_policy" added.
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.3 Configuring the JMS System User for Asynchronous Web Services Using WLST

To configure the JMS system user for asynchronous web services:

1. Attach the `oracle/async_web_service_policy` policy using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   async_web_service_policy")

   Policy reference "oracle/async_web_service_policy" added.
   ```

2. Configure the `jms.access.user` policy configuration property using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/async_web_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   async_web_service_policy','jms.access.user','OracleSystemUser')

   The configuration override property "jms.access.user" having value
   "OracleSystemUser" has been added to the reference to policy with URI "oracle/
   async_web_service_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the asynchronous policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable an asynchronous policy configured at a higher scope, attach the `oracle/no_async_web_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_async_web_service_policy")

Policy reference "oracle/no_async_web_service_policy" added.
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.4 Configuring Reliable Messaging Using WLST

To configure web services reliable messaging using WLST:

1. Attach the `oracle/reliable_messaging_policy` policy using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   reliable_messaging_policy")

   Policy reference "oracle/reliable_messaging_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/reliable_messaging_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   reliable_messaging_policy','acknowledgement.interval','P0DT01S')

   The configuration override property "acknowledgement.interval" having value
   "P0DT01S" has been added to the reference to policy with URI "oracle/
   reliable_messaging_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the web service reliable messaging policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a reliable messaging policy configured at a higher scope, attach the `oracle/no_reliable_messaging_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_reliable_messaging_policy")

Policy reference "oracle/no_reliable_messaging_policy" added.
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.5 Configuring Atomic Transactions Using WLST

To configure web services atomic transactions:

1. Attach the `oracle/atomic_transaction_policy` policy using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   atomic_transaction_policy")

   Policy reference "oracle/atomic_transaction_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/atomic_transaction_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   atomic_transaction_policy','version','WSAT11')

   The configuration override property "version" having value "WSAT11" has been
   added to the reference to policy with URI "oracle/atomic_transaction_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the web service atomic transaction policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable an atomic transaction policy configured at a higher scope, attach the `oracle/no_atomic_transaction_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_atomic_transaction_policy")

Policy reference "oracle/no_atomic_transaction_policy" added.
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.6 Configuring MTOM Using WLST

To configure MTOM on the web service using WLST:

1. Attach the `oracle/wsmtom_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/wsmtom_policy")

   Policy reference "oracle/wsmtom_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/wsmtom_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   wsmtom_policy','reference.priority','10')

   The configuration override property "reference.priority" having value "10" has
   been added to the reference to policy with URI "oracle/wsmtom_policy".
   ```

   For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the web service atomic transaction policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable an MTOM policy configured at a higher scope, attach the `oracle/no_mtom_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/no_mtom_policy")

Policy reference "oracle/no_mtom_policy" added.
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.7 Configuring Fast Infoset Using WLST

To configure Fast Infoset on the web service or client using WLST:

1. Attach the `oracle/fastinfoset_service_policy` or `oracle/fastinfoset_client_policy` policy to the web service or client, respectively, using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   fast_infoset_service_policy")

   Policy reference "oracle/fast_infoset_service_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/fast_infoset_service_policy" and "oracle/fast_infoset_client_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   fast_infoset_client_policy','fast.infoset.content.negotiation','OPTIMISTIC')

   The configuration override property "fast.infoset.content.negotiation" having
   value "OPTIMISTIC" has been added to the reference to policy with URI "oracle/
   fast_infoset_client_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the web service Fast Infoset policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a Fast Infoset policy configured at a higher scope, attach the `oracle/no_fast_infoset_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_fast_infoset_service_policy")

Policy reference "oracle/no_fast_infoset_service_policy" added.
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.8 Configuring SOAP Over JMS Transport Using WLST

To configure SOAP over JMS transport on the web service or client using WLST:

1. Attach the `oracle/jms_transport_service_policy` or `oracle/jms_transport_client_policy` policy to the web service or client, respectively, using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   jms_transport_service_policy")

   Policy reference "oracle/jms_transport_service_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/jms_transport_service_policy" and "oracle/jms_transport_client_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   jms_transport_service_policy','jndi.connection.factory.name','com.oracle.webservi
   ces.jms.ConnectionFactory')

   The configuration override property "jndi.connection.factory.name" having value
   "com.oracle.webservices.jms.ConnectionFactory" has been added to the reference
   to policy with URI "oracle/jms_transport_service_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the SOAP over JMS transport policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a SOAP over JMS transport policy configured at a higher scope, attach the `oracle/no_jms_transport_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_jms_transport_service_policy")

Policy reference "oracle/no_jms_transport_service_policy" added.
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.9 Configuring Persistence Using WLST

To configure the persistence using WLST:

1. Attach the `oracle/persistence_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/persistence_policy")

   Policy reference "oracle/persistence_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/persistence_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   > **Note:**
   >
   > For Java SE clients, you can configure `oracle:jrf:Coherence` only.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   persistence_policy','providerName', 'oracle:jrf:Coherence')

   The configuration override property "peristence_policy" having value
   "oracle:jrf:Coherence" has been added to the reference to policy with URI
   "oracle/persistence_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the persistence policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- To disable a persistence policy configured at a higher scope, attach the `oracle/no_persistent_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

  ```
  wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
  no_persistent_policy")

  Policy reference "oracle/no_persistent_policy" added.
  ```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.10 Enabling or Disabling Web Services Using WLST

When a web service application is deployed, the web service endpoint is enabled by default if no errors are encountered. If there are errors, the web service application is deployed, but the web service endpoint is not enabled.

You may need to temporarily make a web service unavailable by disabling the web service. For example, you may need to correct an invalid policy reference. When you disable a web service, requests to the web service will fail. To disable a web service, you must make the endpoint on which the web service receives requests unavailable.

To enable a web service endpoint using WLST:

1. Attach the `oracle/request_processing_service_policy` policy using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   request_processing_service_policy")

   Policy reference "oracle/request_processing_service_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/request_processing_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride("oracle/
   request_processing_service_policy", "reference.priority", "10")
   ```

   For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To disable the web service access policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a web service access policy configured at a higher scope, attach the `oracle/no_pox_http_binding_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

  ```
  wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
  no_request_processing_service_policy")

  Policy reference "oracle/no_request_processing_service_policy" added.
  ```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.11 Enabling or Disabling Public Access to the Web Service WSDL Document Using WLST

To enable public access to the web service WSDL document using WLST:

1. Attach the `oracle/wsdl_request_processing_service_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   wsdl_request_processing_service_policy")

   Policy reference "oracle/wsdl_request_processing_service_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/wsdl_request_processing_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   wsdl_request_processing_service_policy','reference.priority,'10')

   The configuration override property "reference.priority" having value "10" has
   been added to the reference to policy with URI "oracle/
   wsdl_request_processing_service_policy".
   ```

   For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the WSDL access policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a WSDL access policy configured at a higher scope, attach the `oracle/no_wsdl_request_processing_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

  ```
  wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
  no_wsdl_request_processing_service_policy")

  Policy reference "oracle/no_wsdl_request_processing_service_policy" added.
  ```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.12 Enabling or Disabling the Processing of SOAP Requests Using WLST

To enable the processing of SOAP requests using WLST:

1. Attach the `oracle/soap_request_processing_service_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   soap_request_processing_service_policy")

   Policy reference "oracle/soap_request_processing_service_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/soap_request_processing_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   soap_request_processing_service_policy','reference.priority', '10')

   The configuration override property "reference.priority" having value "10" has
   been added to the reference to policy with URI "oracle/
   soap_request_processing_service_policy".
   ```

   For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

```
wls:/wls_domain/serverConfig> commitWSMSession()

The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
Service({http://ejb.oracle.com/
targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
```

To disable the SOAP request processing policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a SOAP request processing policy configured at a higher scope, attach the `oracle/no_wsdl_request_processing_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

  ```
  wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
  no_soap_request_processing_service_policy")

  Policy reference "oracle/no_soap_request_processing_service_policy" added.
  ```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.13 Enabling or Disabling Non-SOAP XML Message Processing Using WLST

To enable an endpoint to receive non-SOAP XML messages that are processed by a user defined `javax.xml.ws.Provider<T>.invoke` method using WLST:

1. Attach the `oracle/pox_http_binding_service_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   pox_http_binding_service_policy")

   Policy reference "oracle/pox_http_binding_service_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/pox_http_binding_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   pox_http_binding_service_policy','reference.priority,'10')

   The configuration override property "reference.priority" having value "10" has
   been added to the reference to policy with URI "oracle/
   pox_http_binding_service_policy".
   ```

For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

```
wls:/wls_domain/serverConfig> commitWSMSession()

The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
Service({http://ejb.oracle.com/
targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
```

To disable the non-SOAP XML message policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a non-SOAP XML message policy configured at a higher scope, attach the `oracle/no_pox_http_binding_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_pox_http_binding_service_policy")

Policy reference "oracle/no_pox_http_binding_service_policy" added.
```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.14 Setting the Log Level for Diagnostic Logs Using WLST

To set the logging level for diagnostic logs for the web service endpoint using WLST:

1. Attach the `oracle/ws_logging_level_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
ws_logging_level_policy")

Policy reference "oracle/ws_logging_level_policy" added.
```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/ws_logging_level_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

```
wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
ws_logging_level_policy','logging.level', 'INFO')
```

```
The configuration override property "loggging.level" having value "INFO" has
been added to the reference to policy with URI "oracle/ws_logging_level_policy".
```

3. Commit the session using the `commitWSMSession` command, for example:

```
wls:/wls_domain/serverConfig> commitWSMSession()

The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
Service({http://ejb.oracle.com/
targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
```

To disable the log level policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a log level policy configured at a higher scope, attach the `oracle/no_ws_logging_level_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_ws_logging_level_policy")

Policy reference "oracle/no_ws_logging_level_policy" added.
```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.15 Overview of Enabling or Disabling the Web Services Test Client Using WLST

You can enable or disable the Web Services Test Client, as described in "Using the Web Services Test Client", at the domain or web service endpoint level:

- "Enabling or Disabling the Web Services Test Client at the Domain Level Using WLST"
- "Enabling or Disabling the Web Services Test Client at the Web Service Endpoint Level Using WLST"

> **Note:**
>
> The procedures described in this section do not impact the availability of the **Web Services Test** link on the Web Service Endpoint page, which enables you to access the Fusion Middleware Control Test Web Service page. For more information, see "Test Web Service Page in Fusion Middleware Control".

4.2.5.15.1 Enabling or Disabling the Web Services Test Client at the Domain Level Using WLST

To enable or disable the web services Test Client at the domain level, use set the `WebServiceTestEnable` property for the domain to `true` to enable the test client and to `false` to disable it.

For example, to enable the Web Services Test Client at the domain level using WLST:

```
wls:/wls-domain/serverConfig> edit()
Location changed to edit tree. This is a writable tree with
DomainMBean as the root. To make changes you will need to start
an edit session via StartEdit().

For more help, use help('edit')

wls:/mydomain/edit> startEdit()
Starting an edit session ...
Started edit session, please be sure to save and activate
your changes once you are done.
wls:/mydomain/edit> cd('WebserviceTestpage')
wls:/mydomain/edit/WebserviceTestpage !> cd('new_domain')
wls:/mydomain/edit/WebserviceTestpage/new_domain !> set('Enabled','true')
wls:/mydomain/edit/WebserviceTestpage/new_domain !> save()
Saving all your changes ...
Saved all your changes successfully.
wls:/mydomain/edit/WebserviceTestpage/new_domain !> activate()
Activating all your changes, this may take awhile ...
The edit lock associated with the edit session is released
once the activation is complete.
Activation completed
wls:/mydomain/edit/WebserviceTestpage/new_domain !>
```

For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.15.2 Enabling or Disabling the Web Services Test Client at the Web Service Endpoint Level Using WLST

To enable the web services test client at the web service endpoint level using WLST:

1. Attach the `oracle/test_page_processing_service_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   test_page_processing_service_policy")

   Policy reference "oracle/test_page_processing_service_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/test_page_processing_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

```
wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
test_page_processing_service_policy','reference.priority,'10')

The configuration override property "reference.priority" having value "10" has
been added to the reference to policy with URI "oracle/
test_page_processing_service_policy".
```

For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

```
wls:/wls_domain/serverConfig> commitWSMSession()

The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
Service({http://ejb.oracle.com/
targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
```

To disable the Web Services Test Client policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a Web Services Test Client policy configured at a higher scope, attach the `oracle/no_ws_logging_level_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_test_page_processing_service_policy")

Policy reference "oracle/no_test_page_processing_service_policy" added.
```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.16 Enabling or Disabling the Exchange of Metadata Using WLST

To enable the exchange of web service metadata using WLST:

1. Attach the `oracle/mex_request_processing_service_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
mex_request_processing_service_policy")

Policy reference "oracle/mex_request_processing_service_policy" added.
```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/

mex_request_processing_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

```
wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
mex_request_processing_service_policy','reference.priority,'10')

The configuration override property "reference.priority" having value "10" has
been added to the reference to policy with URI "oracle/
mex_request_processing_service_policy".
```

For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

```
wls:/wls_domain/serverConfig> commitWSMSession()

The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
Service({http://ejb.oracle.com/
targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
```

To disable the metadata exchange policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a metadata exchange policy configured at a higher scope, attach the `oracle/no_mex_request_processing_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
no_mex_request_processing_service_policy")

Policy reference "oracle/no_mex_request_processing_service_policy" added.
```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.17 Enabling or Disabling MTOM-encoded SOAP Fault Messages Using WLST

To enable the creation of MTOM-encoded SOAP fault messages when MTOM is enabled using WLST:

1. Attach the `oracle/mtom_encode_fault_service_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

```
wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
mtom_encode_fault_service_policy")

Policy reference "oracle/mtom_encode_fault_service_policy" added.
```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/mtom_encode_fault_service_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   mtom_encode_fault_service_policy','reference.priority,'10')

   The configuration override property "reference.priority" having value "10" has
   been added to the reference to policy with URI "oracle/
   mtom_encode_fault_service_policy".
   ```

   For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the MTOM-encoded fault message policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable an MTOM-encoded fault message policy configured at a higher scope, attach the `oracle/no_mtom_encode_fault_service_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

  ```
  wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
  no_mtom_encode_fault_service_policy")

  Policy reference "oracle/no_mtom_encode)fault_service_policy" added.
  ```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.18 Validating the Request Message Using WLST

To enable the validation of request messages using WLST:

1. Attach the `oracle/schema_validation_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   schema_validation_policy")

   Policy reference "oracle/schema_validation_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/schema_validation_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   schema_validation_policy','reference.priority', '10')

   The configuration override property "reference.priority" having value "10" has
   been added to the reference to policy with URI "oracle/schema_validation_policy".
   ```

   For more information about overriding the `reference.priority` configuration property, see "Specifying the Priority of a Policy Attachment" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the schema validation policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a schema validation policy configured at a higher scope, attach the `oracle/no_schema_validation_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

  ```
  wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
  no_schema_validation_policy")

  Policy reference "oracle/no_schema_validation_policy" added.
  ```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.19 Setting the Maximum Size of the Request Message Using WLST

To set the maximum size of the request message using WLST:

1. Attach the `oracle/max_request_size_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   max_request_size_policy")

   Policy reference "oracle/max_request_size_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/max_request_size_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   max_request_size_policy','max.request.size', '-1')

   The configuration override property "max.request.size" having value "-1" has
   been added to the reference to policy with URI "oracle/max_request_size_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the maximum request size policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a maximum request size policy configured at a higher scope, attach the `oracle/no_max_request_size_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

  ```
  wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
  no_max_request_size_policy")

  Policy reference "oracle/no_max_request_size_policy" added.
  ```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.20 Configuring Binary Caching of Content

To enable and configure the binary caching of content using WLST:

1. Attach the `oracle/cache_binary_content_policy` to the web service using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager.*

   For example:

   ```
   wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
   cache_binary_content_policy")

   Policy reference "oracle/cache_binary_content_policy" added.
   ```

2. Configure the policy using the `setWSMPolicyOverride` command. For a list of configuration properties that you can override, see "oracle/cache_binary_content_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

   ```
   wls:/wls_domain/serverConfig> setWSMPolicyOverride('oracle/
   cache_binary_content_policy','mode',
   'com.oracle.webservices.api.CacheBinaryContentMode.BINARY')

   The configuration override property "mode" having value
   "com.oracle.webservices.api.CacheBinaryContentMode.BINARY" has been added to the
   reference to policy with URI "oracle/cache_binary_content_policy".
   ```

3. Commit the session using the `commitWSMSession` command, for example:

   ```
   wls:/wls_domain/serverConfig> commitWSMSession()

   The policy set for subject "/weblogic/base_domain/jaxwsejb30ws|#jaxwsejb|WS-
   Service({http://ejb.oracle.com/
   targetNamespace}EchoEJBService#EchoEJBServicePort)" was saved successfully.
   ```

To disable the binary content caching policy, perform one of the following steps:

- Detach the policy. For more information, see "Detaching Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Disable the policy. For more information, see "Enabling and Disabling Directly Attached Policies Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- To disable a binary content caching policy configured at a higher scope, attach the `oracle/no_cache_binary_content_policy` using the `attachWSMPolicy` command. For complete details, see "Attaching Policies Directly Using WLST" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

  For example:

  ```
  wls:/wls_domain/serverConfig> attachWSMPolicy("oracle/
  no_cache_binary_content_policy")

  Policy reference "oracle/no_cache_binary_content_policy" added.
  ```

  For more information about the WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

4.2.5.21 Configuring Virtual User Using WLST

To configure a virtual user using WLST:

1. Begin repository session and select token issuer trust document configured with the Domain.

   For example:

```
wls:/wls_domain/serverConfig> beginWSMSession()

Session started for modification.

wls:/wls_domain/serverConfig>selectWSMTokenIssuerTrustDocument("DOMAIN-
WLS-jrfServer_domain")

Token Issuer Trust document named "DOMAIN-WLS-jrfServer_domain" selected in the
session.
```

2. Create Virtual User for DN.

   For example:

```
wls:/wls_domain/
serverConfig>setWSMTokenIssuerTrustVirtualUser("CN=alice","true",
["member"],["urn:dir:attribute-def:personAffiliation"])

New TokenAttributeRule added for DN: CN=alice.
Virtual user created for DN: CN=alice
```

3. Add Token and Mapping Roles

   For example:

```
wls:/wls_domain/
serverConfig>setWSMTokenIssuerTrustVirtualUserRoleMapping("CN=alice","st
aff", ["manager", "executer"])

New token role, staff, has been created with the given mapping values.
```

4. Display Token Attribute Rules.

   For example:

```
wls:/wls_domain/serverConfig>displayWSMTokenIssuerTrustAttributeRule(None)

DN      : CN=weblogic,OU=Orakey Test Encryption Purposes Only,O=Oracle,C=US

DN      : CN=alice

wls:/wls_domain/
serverConfig>displayWSMTokenIssuerTrustAttributeRule("CN=alice")

DN      : CN=alice
List of Other Attributes
            None

Virtual user is enabled.

List of default roles : member

List of token role attribute(s): urn:dir:attribute-def:personAffiliation
```

```
List of token role mapping(s):
            Token Role Name: staff
            List of token mapping role(s): manager, executer
```

Perform the following steps to delete token and mapping roles, disable a virtual user, or delete a virtual user:

- Delete token and mapping roles.

  For example:

  ```
  wls:/wls_domain/serverConfig>
  setWSMTokenIssuerTrustVirtualUserRoleMapping("CN=alice","staff")

  Token role, staff, and its mapping values have been deleted.
  ```

- Disable virtual user.

  For example:

  ```
  wls:/wls_domain/serverConfig>
  setWSMTokenIssuerTrustVirtualUser("CN=alice","false")

  Virtual user updated for DN: CN=alice
  ```

- Delete virtual user.

  For example:

  ```
  wls:/wls_domain/serverConfig>
  deleteWSMTokenIssuerTrustVirtualUser("CN=alice")

  Virtual user deleted for DN: CN=alice
  ```

4.2.6 Configuring Web Service Clients Using WLST

Use the following procedure to configure the web service client endpoint (port) using WLST:

1. Connect to the running instance of WebLogic Server to which the application is deployed as described in "Accessing the Web Services Custom WLST Commands".
2. Use the `listWebServiceClients` WLST command to display a list of the web service clients in your application as described in "Viewing Web Service Clients Using WLST".
3. Use the `listWebServiceClientPorts` command to display the endpoint name and endpoint URL for a web service client.

   ```
   listWebServiceClientPorts(application,moduleOrCompName,moduleType,serviceRefName)
   ```

   For example, to display the endpoint for the service reference `client`:

   ```
   wls:/wls-domain/serverConfig> listWebServiceClientPorts('/base_domain/
   AdminServer/application1#V2.0', 'test1', 'wsconn', 'client')

   HelloWorld_pt serviceWSDLURI=http://namespace/soa-infra/services/default/
   HelloWorld/client?wsdl
   ```

4. Use the `listWebServiceClientStubProperties` command to view the configuration details for a web service client endpoint.

```
listWebServiceClientStubProperties(application, moduleOrCompName, moduleType,
serviceRefName,portInfoName)
```

For example, to view the configuration details for the `HelloWorld_pt`:

```
wls:/wls-domain/serverConfig> listWebServiceClientStubProperties('/base_domain/
AdminServer/application1#V2.0', 'test1','wsconn','client','HelloWorld_pt')

keystore.recipient.alias=A1
saml.issuer.name=B1
user.roles.include=C1
```

Alternatively, you can set the `detail` argument to `true` in the `listWebServiceClients` command to view the configuration details for the endpoint as shown in "Viewing Web Service Clients Using WLST".

5. Do one of the following:

   - Use the `setWebServiceClientStubProperty` command to set or change a single stub property of a web service client endpoint. Specify the property to be set or changed using the `propName` and `propValue` arguments. To remove a property, specify a blank value for the `propValue` argument.

     ```
     setWebServiceClientStubProperty(application,moduleOrCompName,moduleType,
      serviceRefName,portInfoName,propName,[propValue])
     ```

     For example, to change the `keystore.recipient.alias` to `oracle` for the `HelloWorld_pt`, use the following command:

     ```
     wls:/wls-domain/serverConfig> setWebServiceClientStubProperty('/base_domain/
     AdminServer/application1#V2.0',  'test1', 'wsconn', 'client',
     'HelloWorld_pt', 'keystore.recipient.alias', 'oracle')
     ```

   - Use the `setWebServiceClientStubProperties` command to configure the set of properties of a web service client endpoint. Specify the properties to be set or changed using the `properties` argument.

     ```
     setWebServiceClientStubProperties(application, moduleOrCompName,
      moduleType, serviceRefName, portInfoName, properties)
     ```

     This command configures or resets all of the stub properties for the OWSM client security policy attached to the client. Each property that you list in the command is set to the value you specify. If a property that was previously set is not explicitly specified in this command, it is reset to the default for the property. If no default exists, the property is removed.

     For example, to configure atomic transactions for the `TaskReference_pt` SOA reference endpoint of the `default/SimpleRef[1.0]` SOA composite application, use the following command:

     ```
     wls:soainfra/serverConfig>
      setWebServiceClientStubProperties('soa-infra' ,'default/SimpleRef[1.0]',
     'soa' ,'client', 'TaskReference_pt', [("wsat.flowOption","SUPPORTS"),
     ("wsat.Version","DEFAULT")])
     ```

     To verify that the reference is properly configured, enter the following command:

     ```
     wls:soainfra/serverConfig>listWebServiceClients(None, None, true)

         /soainfra/soa_server1/soa-infra:
             compositeName=default/SimpleRef[1.0], moduleType=soa,
     ```

```
serviceRefName=client
                TaskReference_pt
                wsat.version=DEFAULT
                wsat.flowOption=SUPPORTS
```

For more information about the client properties that you can set, see Table 4-3. When specifying these properties, use the format shown in the Property Name column.

You can also set the properties described in "Overview of Policy Configuration Overrides" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

For more information about these WLST commands and their arguments, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

5
Testing Web Services

You can test basic and advanced features of your web service, including security, quality of service (QoS), HTTP header options, and so on. You can also perform stress testing of the security features.
After you have deployed a web service to WebLogic Server, you can use one of the following tools to test your web service:

- Web Services Test Client, as described in "Using the Web Services Test Client" on page 5-1.
- Fusion Middleware Control Test Web Service Page, as described in "Test Web Service Page in Fusion Middleware Control" on page 5-14.

5.1 Using the Web Services Test Client

Only the users with roles as `Admin` or `Deployer` can use the Web Services Test Client to test web services.

Using the Web Services Test Client, these users can:

- Test basic functionality to ensure that the web service was deployed and is operating as expected.
- Test basic authentication security.
- Test advanced features, such as web services addressing, atomic transactions, SOAP Message Transmission Optimization Mechanism (MTOM), Fast Infoset, and Oracle Web Service Manager (OWSM) security policies.

> **Note:**
>
> For web services that use SOAP over Java Messaging Service (JMS) transport as the connection protocol, you can test only basic features.

- View the WSDL for the web service and the imported schemas, if applicable.
- Export and import test cases.
- Configure Java keystores (JKS) for use in testing security.
- Configure and use HTTP proxy settings, as required by your environment.

The following sections describe how to use the Web Services Test Client:

- Invoking the Web Services Test Client
- Selecting the Web Service to Test
- Testing Web Service Operations
- Configuring Basic Test Settings
- Testing Advanced Web Service Features and Security

- Viewing the WSDL and Imported Schemas
- Configuring the Web Services Test Client
- Editing the Input Arguments as XML Source
- Viewing the History
- Exporting and Importing Test Cases
- Enabling and Disabling the Web Services Test Client
- Logging Out of the Web Services Test Client

> **See Also:**
>
> - Default Global Roles

5.1.1 Invoking the Web Services Test Client

You can invoke the Web Services Test Client from any browser, using the web service endpoint, or using the Administration Console.
For more information, refer to the following sections:

- "Invoking the Web Services Test Client From a Browser" on page 5-2
- "Invoking the Web Services Test Client Using the Web Service Endpoint" on page 5-3
- "Invoking the Web Services Test Client Using the Administration Console" on page 5-3

> **Note:**
>
> When you first invoke the Web Services Test Client, there will be a brief delay while the application deploys.
>
> To test web services that use SOAP over JMS transport as the connection protocol, the test page must be invoked from the server on which the web service is deployed.

5.1.1.1 Invoking the Web Services Test Client From a Browser

You can invoke the Web Services Test Client from any browser by entering the following URL:

```
http://host:port/ws_utc
```

where:

- `host` refers to the computer on which WebLogic Server is running.
- `port` refers to the port number on which WebLogic Server is listening (default value is `7001`).

When prompted, enter the login credentials for the Web Services Test Client.

The Web Services Test Client home page is invoked. To select the web service WSDL, see "Selecting the Web Service to Test" on page 5-3.

5.1.1.2 Invoking the Web Services Test Client Using the Web Service Endpoint

You can invoke the Web Services Test Client by navigating to the web service endpoint.

- For an Oracle Infrastructure web service, when you navigate to the web service endpoint, the Web Services Test Client is invoked with the web service WSDL selected.
- For JAX-WS and JAX-RPC web services, an intermediary page is invoked. Click **Test** to invoke the Web Services Test Client with the web service WSDL selected.

When prompted, enter the login credentials for the Web Services Test Client.

To test the web service operations, see "Testing Web Service Operations" on page 5-4.

5.1.1.3 Invoking the Web Services Test Client Using the Administration Console

To test a deployed web service using the Administration Console, follow these steps:

1. Invoke the Administration Console in your browser using the following URL:

   `http://[host]:[port]/console`

   where:

   - `host` refers to the computer on which WebLogic Server is running.
   - `port` refers to the port number on which WebLogic Server is listening (default value is `7001`).
2. Select the web service in the Deployments table that you would like to test. For more information, see "How Web Services Are Displayed in the Administration Console" in *Understanding WebLogic Web Services for Oracle WebLogic Server*.
3. Select the **Testing** tab and click the **Test Point** link next to the web service endpoint that you want to test.
4. When prompted, enter the login credentials for the Web Services Test Client.
5. Follow the procedure described in "Test a Web Service" in *Oracle WebLogic Server Administration Console Online Help*.

The Web Services Test Client is invoked with the web service WSDL selected. To test the web service operations, see "Testing Web Service Operations" on page 5-4.

5.1.2 Selecting the Web Service to Test

From the Web Services Test Client home page, you can select the WSDL corresponding to the web service that you want to test. You can return to this page at anytime by selecting **Choose Another WSDL**.
To select the web service to test:

1. Enter the WSDL in the **Enter WSDL URL** field.
2. To use the configured HTTP proxy, click the **HTTP Proxy** check box.

   For information about configuring the proxy, see "Configuring an HTTP Proxy" on page 5-11.
3. Click **Test**.

Alternatively, you can select a WSDL that was previously loaded from the WSDL list. To toggle the list, click **Show WSDL List** or **Hide WSDL List**.

The web service operations that are available to test are displayed. For more information, see "Testing Web Service Operations" on page 5-4.

5.1.3 Testing Web Service Operations

To test a web service operation:

1. Select the web service operation to test using one of the following methods:
   - Select the operation in the Operations pane.
   - With the web service port folder selected in the Operations pane, click **Test** associated with the web service operation that you want to test.
2. Enter a value for each of the parameters in the Parameters section.

   Required parameters are identified using  .

   For information about editing the XML source, see "Editing the Input Arguments as XML Source" on page 5-12.
3. Configure the basic settings for the test client, such as the endpoint URL, basic authentication settings, encoding and binding types, and whether or not you want to use the configured HTTP proxy. For information, see "Configuring Basic Test Settings" on page 5-4.
4. Test the advanced features of your web service by configuring settings (as applicable), as described in "Testing Advanced Web Service Features and Security" on page 5-5.

   Ensure that the advanced feature settings are compatible with the web service endpoint that you are testing. If they are not, the test will fail and the stack error displays in the Test Results section.

   > **Note:**
   >
   > For web services that use SOAP over JMS transport as the connection protocol, you can test only basic features.

5. Click **Invoke**.

The Test Results section is displayed at the bottom of the content area, displaying the SOAP request and response messages. If the test fails, the stack error information is displayed in the Test Results section.

5.1.4 Configuring Basic Test Settings

Basic test settings for a test client includes Endpoint URL, User Name, Password, Encoding, BindingType, and HTTP Proxy.
You can configure basic settings for the Web Services Test Client, including the username and password for basic authentication, by clicking the **Basic Settings** tab in the Settings section, setting the values defined in Table 5–1, and clicking **Invoke** to invoke the web service.

Table 5-1 Basic Test Client Settings

Setting	Description
Endpoint URL	Override the web service endpoint address.
User Name	Username to use for testing basic authentication. **Note**: You configure the JKS keystores, as described in "Configuring the JKS Keystores" on page 5-11
Password	**Password** to use for testing basic authentication. Note: You configure the JKS keystores, as described in "Configuring the JKS Keystores" on page 5-11
Encoding	Encoding standard. Valid values include: `UTF-8` and `UTF-16`.
BindingType	Binding type. Valid values include: • `SOAP Binding`—Standard SOAP web service. • `REST/HTTP Binding`—SOAP over HTTP using the `invoke()` method of the `javax.xml.ws.Provider<T>` interface.
HTTP Proxy	Flag that specifies whether the HTTP proxy is enabled. You can configure the global HTTP proxy setting, as described in "Configuring an HTTP Proxy" on page 5-11. **Note**: This setting is available in development mode only.

5.1.5 Testing Advanced Web Service Features and Security

Advanced web service features include Addressing, Atomic Transactions, MTOM, Fast Infoset, and OWSM security policies.
For more information on how to test the advanced features of a web service, refer to the following sections:

- Testing Addressing
- Testing Atomic Transactions
- Testing MTOM
- Testing Fast Infoset
- Testing OWSM Security Policies

> **Note:**
>
> For web services that use SOAP over JMS transport as the connection protocol, you can test only basic features.

5.1.5.1 Testing Addressing

WS-Addressing provides a transport-neutral mechanism to address web services and their associated messages. Using WS-Addressing, endpoints are uniquely and unambiguously defined in the SOAP header. For more information, see "Using Web Services Addressing" in *Developing JAX-WS Web Services for Oracle WebLogic Server*.

You can test WS-Addressing, if enabled on the web service, by clicking the **Addressing** tab in the Settings section, setting the values defined in Table 5–2, and clicking **Invoke** to invoke the web service.

Table 5-2 Test Settings for WS-Addressing

Setting	Description
Enabled	Flag that specifies whether WS-Addressing is enabled on the Web Service Test Client.
Version	WS-Addressing version. Valid values include: • `W3C`. For more information, see: `http://www.w3.org/2002/ws/addr/` • `Member Submission`. For more information, see `http://www.w3.org/Submission/ws-addressing/`
ReplyTo	Type of the `ReplyTo` header. Valid values include: `anonymous`, `non-anonymous`, and `Addressing None`.
FaultTo	Type of the FaultTo header. Valid values include: `anonymous`, `non-anonymous`, and `Addressing None`.

5.1.5.2 Testing Atomic Transactions

web services enable interoperability with other external transaction processing systems, such as Websphere, Microsoft .NET, and so on, through the support of the following specifications:

- Web Services Atomic Transaction (WS-AtomicTransaction) Versions 1.0, 1.1, and 1.2: `http://docs.oasis-open.org/ws-tx/wstx-wsat-1.2-spec-cs-01/wstx-wsat-1.2-spec-cs-01.html`
- Web Services Coordination (WS-Coordination) Versions 1.0, 1.1, and 1.2: `http://docs.oasis-open.org/ws-tx/wstx-wscoor-1.2-spec-cs-01/wstx-wscoor-1.2-spec-cs-01.html`

For more information about web services atomic transactions, see "Using Web Services Atomic Transactions" in *Developing JAX-WS Web Services for Oracle WebLogic Server*.

You can test atomic transactions, if enabled on the web service, by clicking the **Atomic Transaction** tab in the Settings section, setting the values defined in Table 5–4, and clicking **Invoke** to invoke the web service.

Table 5-3 Test Settings for Atomic Transactions

Setting	Description
Enabled	Flag that specifies whether atomic transactions are enabled on the Web Service Test Client.
Version	Version of the web services atomic transaction coordination context that is used for the Web Service Test Client. Valid values include: `default`, `wsat`, `wsat11`, and `wsat12`. The `default` value for web service clients is `wsat` (equivalent to WS-AT 1.0).
Transaction Flow Type	Flag that specifies whether the web services atomic transaction coordination context is passed with the transaction flow. For information about setting this value, see "Enabling Web Services Atomic Transactions on Web Services" in *Developing JAX-WS Web Services for Oracle WebLogic Server*.
Action After Invocation	Action required after invocation. Valid values include `commit` or `rollback` the transaction.

5.1.5.3 Testing MTOM

SOAP Message Transmission Optimization Mechanism/XML-binary Optimized Packaging (MTOM/XOP) defines a method for optimizing the transmission of XML data of type `xs:base64Binary` or `xs:hexBinary` in SOAP messages. When the transport protocol is HTTP, Multipurpose Internet Mail Extension (MIME) attachments are used to carry that data while at the same time allowing both the sender and the receiver direct access to the XML data in the SOAP message without having to be aware that any MIME artifacts were used to marshal the `base64Binary` or `hexBinary` data.

For more information about MTOM, see:

- JAX-WS: "Optimizing Binary Data Transmission Using MTOM/XOP" in *Developing JAX-WS Web Services for Oracle WebLogic Server*
- Oracle Infrastructure Web Services: "Using MTOM Encoded Message Attachments" in *Developing Oracle Infrastructure Web Services*.

You can test MTOM, if enabled on the web service, by clicking the **MTOM** tab in the Settings section, setting the values defined in Table 5–5, and clicking **Invoke** to invoke the web service.

Table 5-4 Test Settings for MTOM

Setting	Description
Enabled	Flag that specifies whether MTOM is enabled on the Web Service Test Client.
Threshold	Attachment threshold (in bytes) that specifies whether `xs:binary64` data is sent inline or as an attachment. This value defaults to `0` (all data is sent as an attachment).

5.1.5.4 Testing Fast Infoset

Fast Infoset is a compressed binary encoding format that provides a more efficient serialization than the text-based XML format. For more information about Fast Infoset, see

- JAX-WS: "Optimizing XML Using Fast Infoset" in *Developing JAX-WS Web Services for Oracle WebLogic Server*.
- Oracle Infrastructure Web Services: "Optimizing XML Transmission Using Fast Infoset" in *Developing Oracle Infrastructure Web Services*.

You can test Fast Infoset, if enabled on the web service, by clicking the **Fast Infoset** tab in the Settings section, setting the values defined in Table 5–6, and clicking **Invoke** to invoke the web service.

Table 5-5 Test Settings for Fast Infoset

Setting	Description
Enabled	Flag that specifies whether atomic transactions are enabled on the Web Service Test Client.
Negotiation Type	Negotiation strategy. Valid values include: • `none`—No negotiation strategy. • `optimistic`—Client assumes that Fast Infoset is enabled. • `pessimistic`—Initial client request, Fast Infoset is not enabled. Subsequent requests will use Fast Infoset if enabled on the service. For more information about the content negotiation strategy, see: • JAX-WS: "Configuring the Content Negotiation Strategy" in *Developing JAX-WS Web Services for Oracle WebLogic Server*. • Oracle Infrastructure Web Services: "Configuring the Content Negotiation Strategy" in *Developing Oracle Infrastructure Web Services*.

5.1.5.5 Testing OWSM Security Policies

You can test OWSM security policies by clicking the OWSM tab in the Settings section, setting the values defined in Table 5–8, and clicking **Invoke** to invoke the web service.

> **Note:**
>
> For Java EE web services, this tab is available only if you have OWSM installed.
>
> Only a subset of the OWSM predefined policies is available, and you can configure only the attributes presented for those policies. All of the other policy attributes use only the policy defaults.
>
> This means that copies you made of the predefined policies are not available. It also means that you cannot enable Secure Conversation, change the message security algorithm suite, and so forth.

Table 5-6 Test Settings for OWSM

Setting	Description
Enabled	Flag that specifies whether OWSM policies are enabled on the Web Service Test Client.

Table 5-6 (Cont.) Test Settings for OWSM

Setting	Description
Policies	Select the policies that you want to test on the client by selecting the associated check box. You can test only a subset of security policies. **Note**: For the policy selected, you must configure the required properties below. Otherwise, an exception is thrown.
Username	Username used for basic authentication.
Password	Password used for basic authentication.
Keystore Location	Location of the keystore file. Click **Choose File** to navigate to the file in your local directory, which is typically in *DOMAIN_HOME**domain_name*\config\fmwconfig.
Remote Keystore Location	Location of the remote keystore. If not specified, the keystore local to the server is used.
Keystore Password	Password used for keystore access.
Encryption Key Alias	Alias of the key within the keystore that will be used to decrypt the response from the service. This property is not used in WSS11 policies.
Encryption Key Password	Password for the key within the keystore that will be used for decryption. This property is not used in WSS11 policies.
Signature Key Alias	Alias of the key within the keystore that is used for digital signatures. For WSS11 policies, this property is used for mutual authentication only.
Signature Key Password	Password for the alias of the key within the keystore that is used for digital signatures.
Recipient Key Alias	Alias for the recipient's public key that is used to encrypt type outbound message.
SAML Audience URI	Relying party, as a comma-separated URI. This property accepts wildcards.
SAML Issuer Name	SAML issuer name to use when trying access a service that is protected using SAML mechanism.
Include User Roles	User roles in a SAML assertion.
Attesting Mapping Attribute	Mapping attribute used to represent the attesting entity. Only the DN is currently supported. This attribute is applicable only to sender vouches message protection use cases. It is not applicable to SAML over SSL policies.

5.1.6 Viewing the WSDL and Imported Schemas

To view the WSDL for the current web service, click **WSDL**.

To view the imported WSDL and schemas, click **Imported WSDL and Schema**. From the dialog box, click the imported file that you wish to view. Click **x** in the upper right corner to close the dialog box.

5.1.7 Configuring the Web Services Test Client

You can configure the Web Services Test Client to define an HTTP proxy, set the default working directory, or define a Java Keystore (JKS).

> **Note:**
>
> Configuration settings are available in development only.

For more information on how to configure web services, refer to the following sections:

- Configuring an HTTP Proxy
- Configuring the JKS Keystores
- Configuring the Default Working Directory

5.1.7.1 Configuring an HTTP Proxy

You configure an HTTP proxy for your Web Services Test Client on the General Settings page.

To configure an HTTP proxy:

1. Click  in the upper-right corner of the Web Services Test Client.
2. Click **General** in the navigation pane.
3. Enter the proxy host in the **Http Proxy Host** field.
4. Enter the proxy port in the **Http Proxy Port** field.
5. Click **Submit**.

5.1.7.2 Configuring the JKS Keystores

You configure the JKS keystores that are associated with the OWSM security policies that you are testing on the Web Services Test Client Security Settings page.

For more information about defining the JKS keystores on WebLogic Server, see "How to Configure a JKS Keystore on WebLogic Server" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To configure the Java Keystores (JKS):

1. Click  in the upper-right corner of the Web Services Test Client.
2. Click **Security** in the navigation pane.
3. To add a new JKS keystore:
   a. Click **Add**.
   b. Enter a name for the JKS keystore in the **Setting Name** field.

c. Enter the password for the JKS keystore in the **Keystore Password** field.

> **Note:**
>
> Defining the password for the JKS keystore in a production environment is not recommended.

d. Enter the path to the file in the **Keystore File** field, or click **Choose File** to navigate to the file in your local directory.

e. Click **Submit**.

4. To edit or delete an existing JKS keystore:

a. Click **Edit**.

b. To delete a JKS keystore, click  .

c. To edit a JKS keystore, click  , edit the fields, and click **Submit Edit**.

d. To exit edit mode, click **Cancel Edit**.

5.1.7.3 Configuring the Default Working Directory

You can configure the default working directory for the Web Services Test Client. By default, the working directory is set to the following subdirectory within the domain directory:

```
<domain-directory>/tmp/WSTestPageWorkDir
```

To configure the default working directory:

1. Click  in the upper-right corner of the Web Services Test Client.
2. Click **General** in the navigation pane.
3. Edit the **Current Working Home** field to reflect the desired working home directory.
4. Click **Submit**.

5.1.8 Editing the Input Arguments as XML Source

You can view the input arguments in a user-friendly form, or you can edit the XML source code directly.
If you edit the XML source directly, you must enter valid XML. To view the input argument as XML, click **Raw Message**. To toggle back to the user-friendly form, click **Form Entry**.

5.1.9 Viewing the History

You can access the details of the test operations that you have performed from the Invocation History pane.

> **Note:**
>
> The Invocation History pane displays only after you invoke your first test operation and is available in development mode only.

You can view the results for tests executed previously within the current session by clicking on the operation in the Invocation History pane. Failed test instances appear in red in the Invocation History pane.

5.1.10 Exporting and Importing Test Cases

You can export individual test cases from the Web Services Test Client and then import them into another test environment.

> **Note:**
>
> This feature is available in development mode only.

For more information, refer to the following sections:

- Exporting a Test Case
- Importing a Test Case

5.1.10.1 Exporting a Test Case

To export a test case, click  in the upper-right corner of the Web Services Test Client. The test case is saved as an XML file using the following filename: `ws-testcase.xml`. If you save multiple test cases, a suffix is added to the filename as follows: `ws-testcase(n).xml`, where *n* is incremented each time a new test case is saved.

5.1.10.2 Importing a Test Case

To import a text case:

1. Click  in the upper-right corner of the Web Services Test Client.
2. Click **Choose File**.
3. Navigate to the test case file, and click **Open**.

   Click **Import**.

5.1.11 Enabling and Disabling the Web Services Test Client

In a development environment, the Web Services Test Client is enabled, by default. In a production environment, the Web Services Test Client is disabled (and undeployed), by default.
You can enable or disable the Web Services Test Client in one of the following ways:

- Using the Administration Console, as described below.
- Using Fusion Middleware Control at the domain or web service endpoint level, as described in "Enabling or Disabling the Web Services Test Client Using Fusion Middleware Control" on page 4-25.
- Using WLST, as described in "Enabling or Disabling the Web Services Test Client Using WLST" on page 4-58.

You should disable the Web Services Test Client to increase security by reducing the externally visible details of an application that exposes web services.

> **Note:**
>
> It is recommended that you *not* enable the Web Services Test Client in production mode. For information about production mode, see "Domain Modes" in *Understanding Domain Configuration for Oracle WebLogic Server*.

To enable or disable the Web Services Test Client at the domain level using the Administration Console:

1. Invoke the Administration Console, as described in "Accessing Oracle WebLogic Administration Console" on page 2-2.
2. Click **Domain** in the Domain Configurations section of the console home page.
3. Select **Configuration > General** to display the general configuration options for the domain.
4. Click **Advanced** to view the advanced configuration settings.
5. Toggle the **Enable Web Service Test Page** configuration flag to enable or disable the Web Services Test Client.
6. Click **Save**.
7. Restart the server so that the configuration settings take effect.

   When enabling the Web Service Test Client in a production environment, the client will be deployed when you restart the server.

5.1.12 Logging Out of the Web Services Test Client

You can log out of the Web Services Test Client at any time by clicking  in the upper-right corner of the Web Services Test Client.

5.2 Test Web Service Page in Fusion Middleware Control

You can use the Test Web Service page in Fusion Middleware Control to verify that you are receiving the expected results from a SOAP web service (WSDL) or a RESTful (WADL) service.
Using the Test Web Service page, you can:

- Test basic functionality to ensure that the web service was deployed and is operating as expected.
- Test advanced features, such as web services addressing, reliable messaging, MTOM, and HTTP headers.
- Test security features.
- Stress test the web service endpoint.

The Test Web Service page allows you to test any of the operations exposed by a WSDL or WADL document. You can test web services that are deployed on any accessible host; the web service does not have to be deployed on this host.

> **Note:**
>
> The Test Web Service page can parse WSDL or WADL URLs that contain ASCII characters only. If the URL contains non-ASCII characters, the parse operation fails. To test a web service that has non-ASCII characters in the URL, allow your browser to convert the WSDL or WADL URL and use the resulting encoded WSDL or WADL URL in the Test Web Service page.
>
> When testing web services that use policies, the OWSM component must be installed in the same domain from which Fusion Middleware Control is being run. Otherwise, an invalid policy exception will be returned.

For more information, refer to the following sections:

- Accessing a Web Service for Testing
- Testing a SOAP Web Service
- Testing an Asynchronous Web Service
- Introduction to Testing a RESTful Web Service
- Testing Security
- Enabling Quality of Service Testing
- Testing HTTP Headers
- Editing the Input Arguments as XML Source
- Stress Testing the Web Service Operation

5.2.1 Accessing a Web Service for Testing

You can navigate to the Test Web Service page in many ways.

The following sections describes two typical ways to access a web service.

Access your web service from the Oracle WebLogic Server Domain Home page

1. In the navigation pane, expand **WebLogic Domain** to show the domain in which you want to test a web service.
2. Select the domain.
3. From the **WebLogic Domain** menu, select **Web Services**, and then **Test Web Service**. The Test Web Service input page appears.
4. Enter the WSDL or WADL URL of the web service you want to test. If you do not know the WSDL or WADL, click the search icon and select from the registered web services, if any.
5. Click **Parse WSDL or WADL**.

   If the WSDL or WADL is secured with HTTP Basic Authentication, click **HTTP Basic Auth Option for WSDL or WADL Access** and enter the username and password before parsing the WSDL or WADL.

   The complete Test Web Services page displays, as shown in Figure 5–1 for a WSDL and in Figure 5–8 for a WADL.

Access your web service from the Web Service Application Home page

1. In the navigation pane, expand **Application Deployments** to view the applications in the domain.
2. Select the application for which you want to test the web service.
3. In the Web Services section of the page, click **Test** for the web service endpoint you want to test.

   The Test Web Service page displays, as shown in Figure 5–1 for a WSDL and in Figure 5–8 for a WADL. Note that the WSDL or WADL field is automatically populated with the WSDL or WADL for the endpoint.

5.2.2 Testing a SOAP Web Service

When the Test Web Services page refreshes with the WSDL details for the SOAP web service, you can select the Service, Port, and Operation type that you want to test, as well as specify a variety of input parameters.

To test a SOAP Web Service:

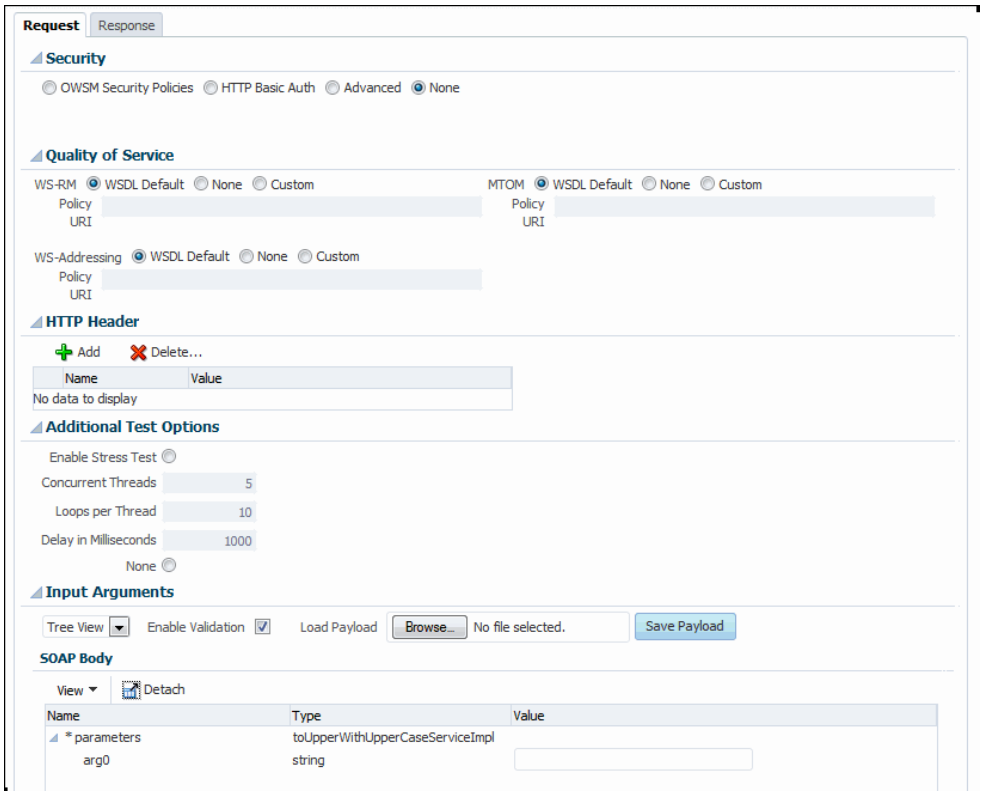
1. Access the Test Web Service page using one of the methods described in Section;5.3.1, "Accessing a Web Service for Testing."
2. Click **Parse WSDL or WADL**.

   If the WSDL is secured with HTTP Basic Authentication, click **HTTP Basic Auth Option for WSDL or WADL Access** and enter the username and password before parsing the WSDL.

Figure 5-1 Test Web Service Page for a WSDL – Collapsed View

3. Select the service and port to be tested. As shown in Figure;5–1, if the WSDL has multiple services and ports, these fields are available as drop-down menus. If the WSDL has only one service and port, these fields are read-only.
4. Select the operation that you want to test from the Operation menu. The available operations are determined from the WSDL.
5. If you want to change the endpoint URL of the test, click **Edit Endpoint URL** and edit the URL.
6. Select the **Request** tab if it is not already selected.
7. Expand the test option sections. The expanded view of the Test Web Service page is shown in Figure;5–2.

Figure 5-2 Bottom Portion of Test Web Service Page for a WSDL – Expanded View



8. In the Security section, specify whether you want to test a web service using OWSM security policies, basic HTTP authentication, authentication from a custom policy, or no credentials. The security setting is not determined from a policy in the WSDL; you can specify the type of security you want to test. The default is **None**. Depending on the option selected, additional fields are displayed. For details about the options available, see Section;5.3.5, "Testing Security."
9. In the Quality of Service section, specify whether you want to explicitly test a Reliable Messaging (WS-RM), WS-Addressing, or MTOM policy. For details about the options available, see "Enabling Quality of Service Testing" on page;5-37.
10. In the HTTP Header section, you can add, modify, or delete HTTP headers to pass request information to a web service. For more information, see "Testing HTTP Headers" on page;5-37.
11. In the Additional Test Options section, select the **Enable Stress Test** option if you want to invoke the web service multiple times simultaneously. If you select this option, you can also provide values for the stress test options, or accept the defaults. For more information, see "Stress Testing the Web Service Operation" on page;5-38.

> **Note:**
>
> The Asynchronous Test Response and Response Key fields are unique for testing asynchronous web services. For more information, see Section;5.3.3, "Testing an Asynchronous Web Service".

12. In the Input Arguments section, enter the input arguments for the web service in the **Value** fields. The parameters and type, and the required input values, are determined from the WSDL.

    Select **Tree View** or **XML View** to toggle between a hierarchical list of input parameters and the XML content.

    For SOAP web services, you can save the payload as XML and can load a previously saved payload. Click **Save Payload** to save the current payload as an XML file on your local file system. To load a saved payload, click **Browse** to find and select one, or click **Update** to replace the currently loaded payload. (**Browse** is replaced by **Update** when a saved payload is loaded.)

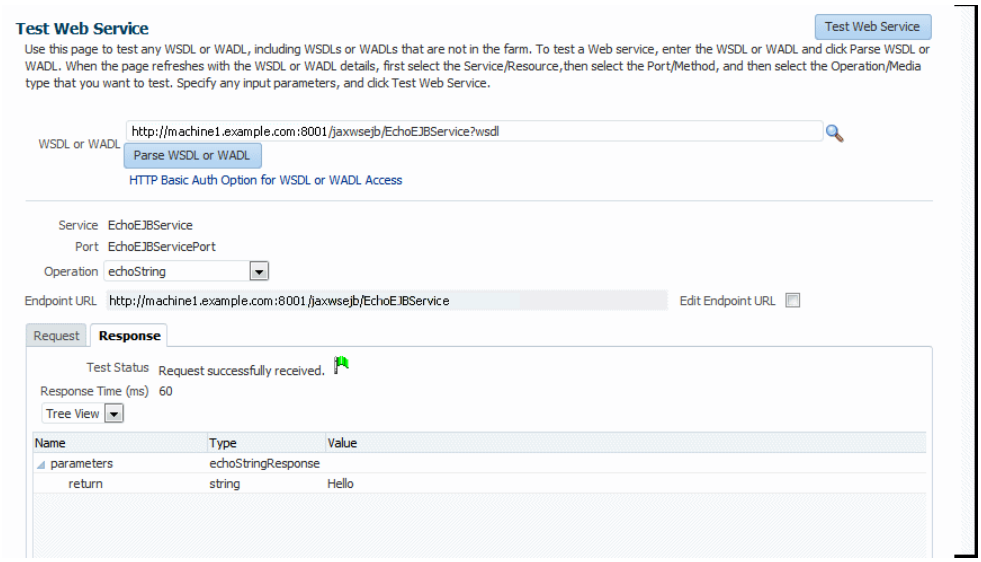
13. Click **Test Web Service** to initiate the test.

    The test results appear in the **Response** tab upon completion.

    If the test is successful, the **Test Status** field indicates *Request Successfully received* and the response time is displayed, as shown in Figure;5–3.

> **Note:**
>
> When running SOA composite tests, the Response tab will indicate whether a new composite was generated. You can also click the **Launch Flow Trace** button to open the Flow Trace window, where you can view the flow of the message through various composite and component instances.

Figure 5-3 Successful Test for a WSDL

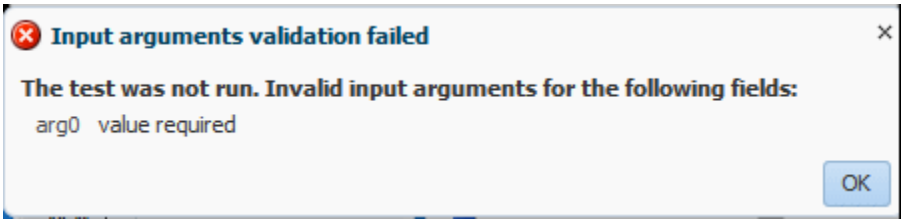


If the test fails, an error message is displayed. For example, Figure;5–4 shows an error resulting from a type error in the *var-Int* parameter. In this particular instance, *string* data was entered when an *int* was expected.

> **Note:**
>
> The results on the **Response** tab are a simplified version of the standard web service results.

14.

Figure 5-4 Data Validation Error

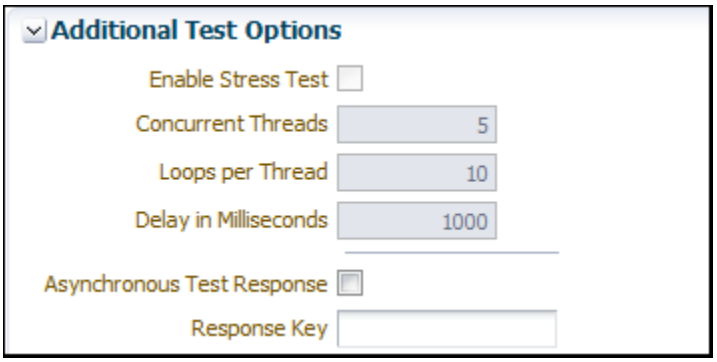


5.2.3 Testing an Asynchronous Web Service

When the Test Web Services page refreshes with the WSDL details for an asynchronous web service, it is nearly identical to the Test Web Services page for a SOAP web service, except that Additional Test Options are available.

- Asynchronous Test Response – Specifies whether to request an asynchronous response upon running the test.
- Response Key – Enter a value (for example, *myasynctest*), which will correlate with the test's asynchronous response. Each response key must be unique. The asynchronous response can also be tracked later on the Asynchronous Test Response page by specifying the corresponding response key.

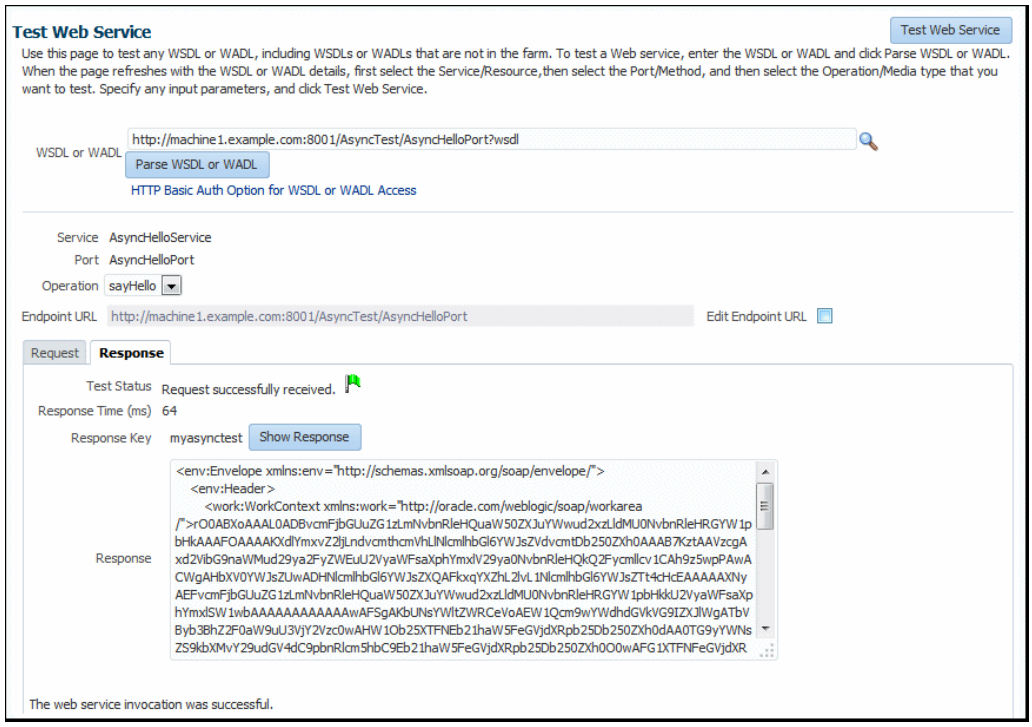
Figure 5-5 Asynchronous Testing Options on the Test Web Service Page



If the test is successful, the Test Status field indicates *Request Successfully received* with the response time. The user-defined Response Key value that corresponds with the asynchronous response is also displayed.

Click **Show Response** to display the test response, if a response exists, in the Response field in XML format, as shown in Figure 5–6.

Figure 5-6 Successful Test and Response for an Asynchronous Web Service



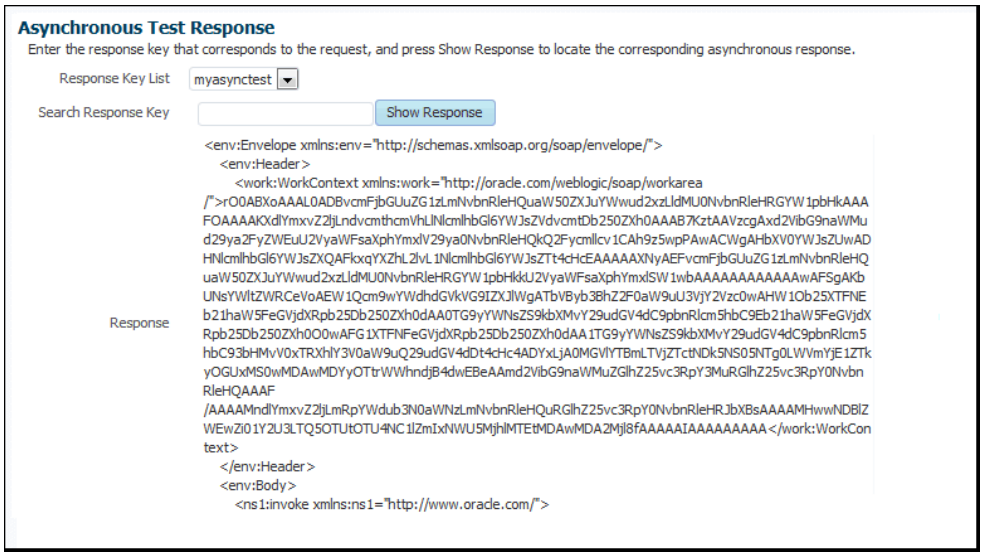
> **Note:**
>
> An asynchronous request can only have a response if the server has not been restarted.

If the response is not available, it can also be tracked later on the Asynchronous Test Response page by using the corresponding response key.

Accessing an asynchronous test response after testing

1. In the navigation pane, expand **WebLogic Domain** to show the domain in which you want to access an asynchronous web service test response.
2. Select the domain.
3. From the **WebLogic Domain** menu, select **Web Services**, and then **Asynchronous Test Response**.
4. On the Asynchronous Test Response page, specify the corresponding response key by either:
   - Using the Response Key List to select an existing, user-defined response key.
   - Using Search Response Key field to search for an existing, user-defined response key.
5. Click **Show Response** to display the test response, if a response exists, in the Response field in XML format.

Figure 5-7 Asynchronous Test Response page showing a test response



6. Repeat step 4 to access and view any other asynchronous test responses that have a response key.

5.2.4 Introduction to Testing a RESTful Web Service

The Web Application Description Language (WADL) is an XML-based file format that describes a RESTful web services application.

When the Test Web Services page refreshes with the WADL details for the RESTful web service, it is similar to the Test Web Services page for a WSDL, except that you can select the Resource, Method, and Media type that you want to test. You can also specify a variety of input parameters, though for this release they are more limited compared to a WSDL page.

- Testing a RESTful Web Service
- The WebRestApp1 Web Service
- The Mime_Multipart_rs Web Service

5.2.4.1 Testing a RESTful Web Service

Access the Test Web Service page to test RESTful web services.

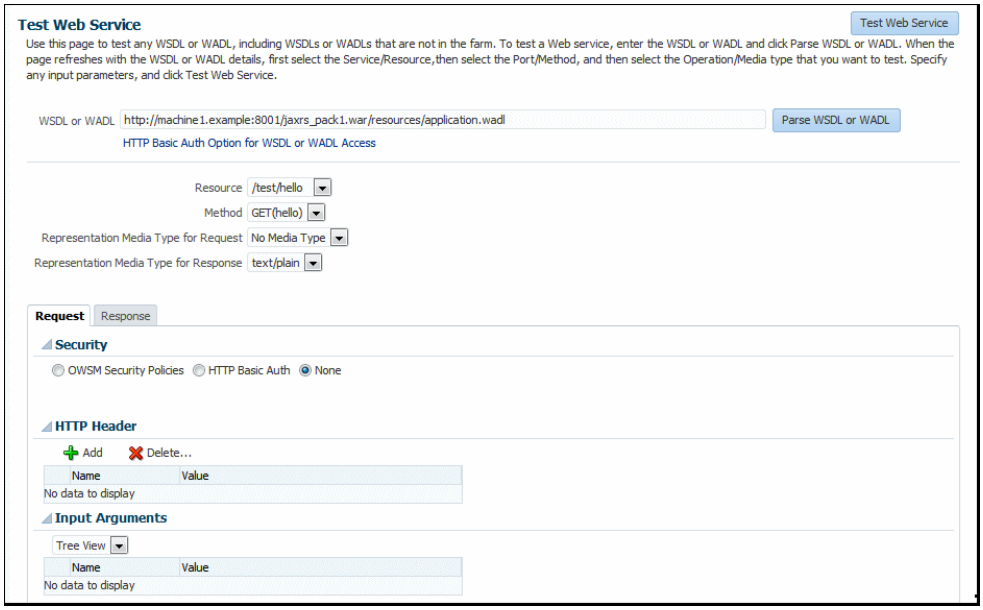
To test a RESTful web service:

1. Access the Test Web Service page using one of the methods described in Section 5.3.1, "Accessing a Web Service for Testing."

2. Click **Parse WSDL or WADL**.

   If the WADL is secured with HTTP Basic Authentication, click **HTTP Basic Auth Option for WSDL or WADL Access** and enter the username and password before parsing the WADL.

Figure 5-8 Test Web Service Page for a WADL



3. Select the resource and method type to be tested. The Method field displays the corresponding HTTP method for the selected resource path in the Resource field

   If the WADL has multiple resources methods, these fields are available as drop-down menus, as shown in Figure;5–8. If the WADL has only one resource and method, these fields are read-only.

> **Note:**
>
> Only the GET and POST methods are supported in this release. (Other methods are ignored if they are present in the WADL.)

4. Select the MIME media type that you want to use for the *test request* from the Representation Media Type for Request menu. The available types, if one exists, are determined from the WADL.
5. Select the MIME media type that you want to use for the *test response* from the Representation Media Type for Response menu. The available types, if one exists, are determined from the WADL. The following media types are supported in this release:
   - application/xml
   - application/json
   - text/plain
   - text/xml

- application/x-www-form-urlencoded

6. Select the **Request** tab if it is not already selected.
7. In the Security section, the options are **OWSM Security Policies**, **HTTP Basic Authentication**, and **None**. The default is **None**. For more information, see "Testing Security" on page;5-31.
8. In the HTTP Header section you can add, modify, or delete HTTP headers to pass request information to a RESTful web service. For more information, see "Testing HTTP Headers" on page;5-37.
9. In the Input Arguments section, enter the input arguments for the RESTful web service in the **Value** fields. The parameters and type and the required input values, are determined from the WADL. The **Style** column enables you to visualize REST services with input parameters of the following, different styles:
   - **Template:** Parameters that parameterize the path to a REST request. (RESOURCE level only).
   - **Matrix:** Parameters that define parameters to be added to the actual path of the resource, but preceding the query string.
   - **Query:** Parameters that are appended to the path of the URL when submitting a request (most common parameter type). For example, you typically see them added to the path after the '?' in a GET HTTP request or as a Key value pair in the request body of the POST and PUT request as part of HTML form submit.
   - **Header:** Parameters that are added to the out-going headers of a HTTP request.

   Style values are determined by your **Resource** and **Method** menu selections.

   Select **Tree View** or **Raw View** to toggle between a hierarchical list of input parameters and the raw data view, where you can provide input directly to the request Header and Body sections.

   > **Note:**
   >
   > **Tree View** is available only for GET methods.

10. Click **Test Web Service** to initiate the test.

    The test results appear in the **Response** tab upon completion. A message and an icon indicates the status of the response: a green flag for success, a red flag for failure, and a yellow flag for not yet executed.

    If the test is successful, the **Test Status** field indicates *Request Successfully received*, with the green flag, and the response time is displayed, as shown in Figure;5–9.

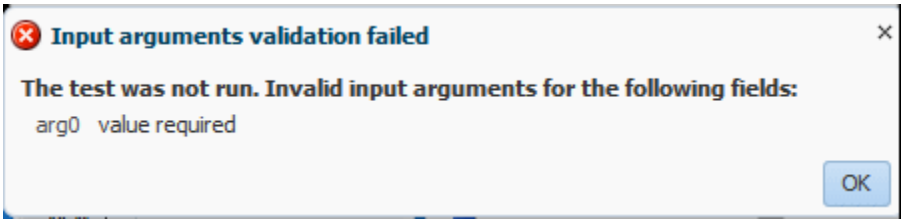
Figure 5-9 Successful Test for a WADL

Only the **Raw View** is available on the Response tab, which will display the entire response from the request invocation.

If the test fails, an error message is displayed. For example, Figure;5–10 shows an error resulting from a type error.

11.

Figure 5-10 Data Validation Error



5.2.4.2 The WebRestApp1 Web Service

You can view and manage token issuer trust configurations using a set of representational state transfer (REST) resources.

The following table lists the REST Resources for WebRestApp1.

Resource	Method	Resource Path
persons	GET	rest/persons
headerGetPerson	GET	rest/headerGetPerson
person{id}	GET	rest/person/{id}
personMatrixAdd	POST	rest/personMatrixAdd
personXXXAdd	POST	rest/personXXXAdd
personPathAdd	POST	rest/personPathAdd/{id}
headerAddPerson	POST	rest/headerAddPerson

Resource	Method	Resource Path
personQueryAdd	POST	rest/personQueryAdd
personAddForm	POST	rest/personAddForm
personQueryDefaultAdd	POST	rest/personQueryDefaultAdd
persondelete	DELETE	rest/persondelete
personUpdate	PUT	rest/personUpdate
verifyPart1	POST	rest/verifyPart1

5.2.4.2.1 GET persons Method

Use the GET method to view all created persons.

REST Request

```
GET rest/persons
```

Input Parameters

None.

Response

Media Type	Result
application/xml	first_name, last_name, ph_no and ID of all available persons created, in XML format
application/json	first_name, last_name, ph_no and ID of all available persons created, in Json format

5.2.4.2.2 GET headerGetPerson Method

Use the GET method to view the header details for a person with specified firstname.

REST Request

```
GET rest/headerGetPerson
```

Input Parameters

Create a HTTP header with name/value as “fname/<firstname>".

Response

Media Type	Result
application/xml	first_name, last_name, ph_no and ID of thefname entered in header, in XML format
application/json	first_name, last_name, ph_no and ID of thefname entered in header, in Json format

5.2.4.2.3 GET person{id} Method

Use the GET method to view a person with specified ID.

REST Request

```
GET rest/person/{id}
```

Input Parameters

Enter the id of the person to be searched.

Response

Media Type	Result
application/xml	all the details of ID entered in inputparameters, in XML format
application/json	all the details of ID entered in inputparameters, in Json format

5.2.4.2.4 POST personMatrixAdd Method

Use the POST method to create a new person and details.

REST Request

```
POST rest/personMatrixAdd
```

Input Parameters

Enter string values for FirstName and LastName, Integer values for ID and Phone_no.

Response

Media Type	Result
application/xml	first_name, last_name, ph_no and ID ofperson created, in XML format
application/json	first_name, last_name, ph_no and ID ofperson created, in Json format

5.2.4.2.5 POST personXXXAdd Method

Use the POST method to create a person's details.

REST Request

```
POST rest/personXXXAdd
```

Input Parameters

Input parameters depend on the media type:

- When selecting application/vnd.xxx+xml for request media type:

  ```
  <person> <firstname>Firstname 12345/<firstname> <id>12345/id
  <lastname>Last 12345/lastname <phone_no>12345/phone_no/<person>
  ```

- When selecting application/vnd.xxx+json for request media type:
  ```
  {"firstname":"cc","id":14,"lastname":"cc","phone_no":14} <person>
  ```

Response

Text message to confirm person has been added.

5.2.4.2.6 POST personPathAdd Method

Use the POST method to create a person with details.

REST Request

```
POST rest/personPathAdd/{id}/{fname}/{lname}/{phno}
```

Input Parameters

Enter string values for FirstName and LastName, Integer values for ID and Phone_no.

Response

Text message to confirm person has been added.

5.2.4.2.7 POST headerAddPerson Method

Use the POST method to create an HTTP header with person details.

REST Request

POST rest/headerAddPerson

Input Parameters

Create a HTTP header with name/value as:

```
“fname = <firstname>"
```

```
“lname = <lastname>"
```

```
“phone_no = <phone_no>"
```

```
“id = <id>"
```

Response

Media Type	Result
application/xml	first_name, last_name, ph_no and ID ofperson created in XML format
application/json	first_name, last_name, ph_no and ID ofperson created in Json format

Text message to confirm person has been added.

5.2.4.2.8 POST personQueryAdd Method

Use the POST method to create a person with details.

REST Request

```
POST rest/personQueryAdd
```

Input Parameters

Enter string values for FirstName and LastName, Integer values for ID and Phone_no.

Response

Text message to confirm person has been added.

5.2.4.2.9 POST personAddForm Method

Use the POST method to create a person with details.

REST Request

```
POST rest/personAddForm
```

Input Parameters

Enter string values for FirstName and LastName, Integer values for ID and Phone_no.

Response

Text message to confirm person has been added.

5.2.4.2.10 POST personQueryDefaultAdd Method

Use the POST method to create a new person and details.

REST Request

```
POST rest/personQueryDefaultAdd
```

Input Parameters

This method may be invoked with or without input parameters specified.

- Enter string values for FirstName and LastName, Integer values for ID and Phone_no.
- Keep all the values blank and invoke the test; default values are picked up and person is created

Response

Media Type	Result
application/xml	first_name, last_name, ph_no and ID ofperson created, in XML format
application/json	first_name, last_name, ph_no and ID ofperson created, in Json format

5.2.4.2.11 DELETE persondelete Method

Use the DELETE method to delete a specified person.

REST Request

```
DELETE rest/persondelete
```

Input Parameters

Enter the id of the person to be deleted.

Response

Text message to confirm person has been deleted.

5.2.4.2.12 PUT personUpdate Method

Use the PUT method to update details for a specified person.

REST Request

```
PUT rest/personUpdate
```

Input Parameters

Enter the id of the person to be updated and make changes to other input variables.

Response

Text message to confirm person has been deleted.

5.2.4.2.13 POST verifyPart1 Method

Use the POST method to send content as parts.

REST Request

```
POST rest/verifyPart1
```

Input Parameters

Create PARTs under Control, then Actions, as follows:

1. Enter part name as “part1" and part content as “part1_contents". Click OK.

2. Select File from Input mode. Enter part name as “part2" and browse for any file. Click OK.

Response

Result contains the number of parts sent and details about each part.

5.2.4.3 The Mime_Multipart_rs Web Service

You can view and manage greeting string multipart configurations using a set of representational state transfer (REST) resources, as summarized below.

Resource	Method	Resource Path
greetingsMultipart	POST	/hello/greetingsMultipart
greetings	GET	/hello/{greetings}
greetingsImagePOST	POST	/hello/greetingsImagePOST
greetings_matrix_to	GET	/hello/greetings_matrix_to
greetings_query_to	GET	/hello/greetings_query_to
greetingsMsg	GET	/hello/greetingsMsg
greetingsImagePOST1	POST	/hello/greetingsImagePOST1
greetingsMsgQP	POST	/hello/greetingsMsgQP
hello	GET	/hello

5.2.4.3.1 POST greetingsMultipart Method

Use the POST method to send content as parts.

REST Request

```
POST /hello/greetingsMultipart
```

Input Parameters

Create PARTs under Control, then Actions, as follows:

1. Enter part name as “part1" and part content as “part1_contents". Click OK.
2. Select File from Input mode. Enter part name as “part2" and browse for any file. Click OK.

Response

Click the “Click the download response content" button. The text file shows the response.

5.2.4.3.2 GET greetings Method

Use the GET method to view a greeting string.

REST Request

```
GET /hello/{greetings}
```

Input Parameters

Enter a string value for greetings.

Response

Click the “Click the download response content" button. The text file shows the response with the contents of the input you entered.

5.2.4.3.3 POST greetingsImagePOST Method

Use the POST method to add an image file.

REST Request

```
POST /hello/greetingsImagePOST
```

Input Parameters

Browse and add a `.png` image file.

Response

Click the “Click the download response content" button. The text file shows the response with the image file contents.

5.2.4.3.4 GET greetings_matrix_to Method

Use the GET method to view a greeting string.

REST Request

```
GET /hello/greetings_matrix_to
```

Input Parameters

Enter a string value for greetings.

Response

Click the “Click the download response content" button. The text file shows the response with the contents of the input you entered.

5.2.4.3.5 GET greetings_query_to Method

Use the GET method to view a greeting string.

REST Request

```
GET /hello/greetings_query_to
```

Input Parameters

Enter a string value for greetings.

Response

Click the “Click the download response content" button. The text file shows the response with the contents of the input you entered.

5.2.4.3.6 GET greetingsMsg Method

Use the GET method to view a greeting string.

REST Request

```
GET /hello/greetingsMsg
```

Input Parameters

Enter a string value for greetings in the contents text area.

Response

The response shows the contents of the input you entered.

5.2.4.3.7 POST greetingsImagePOST1 Method

Use the POST method to add an image file.

REST Request

```
POST /hello/greetingsImagePOST1
```

Input Parameters

Browse folders and add an image file.

Response

Verify that the attached file size is returned.

5.2.4.3.8 POST greetingsMsgQP Method

Use the POST method to add an image file.

REST Request

```
POST /hello/greetingsMsgQP
```

Input Parameters

Enter a string value for greetings.

Response

Click the “Click the download response content" button. The text file shows the response with the image file contents.

5.2.4.3.9 GET hello Method

Use the GET method to view a greeting string.

REST Request

```
GET /hello
```

Input Parameters

None.

Response

Click the “Click the download response content" button. The text file contains the response.

5.2.5 Testing Security

You can use the Test Web Service Page to test web services security using OWSM security policies, HTTP basic authentication, or custom policies including copies of predefined policies. You can choose the type of test by selecting one of the options in the Security section of the page.

The security setting is not determined from a policy in the WSDL or WADL; you can specify the type of security you want to test. The default is **None**.

The following options are available. They are described in more detail in subsequent sections:

- **OWSM Security Policies**– Uses the credentials and other security options required by the OWSM security policies for authentication and message protection.

  The options available when you select **OWSM Security Policies** differ slightly, depending on whether you are testing a RESTful web service or a SOAP web service, as described later in this section.

- **HTTP Basic Auth** – Inserts the username and password credentials in the HTTP transport header. Both the username and password are required.

  If you do specify a username and password, they must exist and be valid.

- **Advanced** – Uses a custom policy to authenticate the user. All user defined or non-default OWSM policies, and the copies of the default policies are termed as custom policies. You must specify the URI for the policy. You can also specify configuration overrides.

> **Note:**
>
> This option is not available for RESTful web services.

- **None** – No credentials are included.

5.2.5.1 OWSM Security Policies

Only a subset of the OWSM predefined policies is available, and you can configure only the attributes presented for those policies. All of the other policy attributes use the policy defaults. For example, you cannot enable Secure Conversation, change the message security algorithm suite, and so forth.

> **Note:**
>
> For a list of client policies supported by the test function, see "Supported Client Security Policies for SOAP Services" on page 5-36 and "Supported Client Security Policies for RESTful Services" on page 5-37.)

Non-security policies and policies not supported by the test function are shown as read-only and cannot be selected. If a service has service policies whose corresponding client policies are not supported, including copies, those policies are shown as read only. This means that copies you made of the predefined policies are not directly listed and are available only through the Advanced option.

For example, assume that your web service has a copy of the oracle/wss11_message_protection_service_policy policy attached, and you have made a copy of the oracle/wss11_message_protection_client_policy. The test client displays both the oracle/wss11_message_protection_client_policy and oracle/wss11_message_protection_client_policy_copy policies, but the oracle/wss11_message_protection_client_policy_copy policy is unavailable and grayed out.

You can choose one of two approaches:

- If for testing purposes the predefined oracle/wss11_message_protection_client_policy policy satisfies your requirements, select the oracle/wss11_message_protection_client_policy.

  The **JKS Keystore Location** and **JKS Keystore Password** fields are required and presented on the page. Enter the location and password and click **Load Keys**.

- If you want to use your own oracle/wss11_message_protection_client_policy_copy policy:

  1. Select **Advanced**.
  2. Enter the policy URI, in this case oracle/wss11_message_protection_client_policy_copy.
  3. For message protection and SSL policies, the **JKS Keystore Location** and **JKS Keystore Password** properties are required.

     Enter the JKS keystore location and password values using the full property names from "Client Policy Configuration Properties That Can Be Overridden at

Design Time" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*:

```
oracle.wsm.security.util.SecurityConstants.ClientConstants.WSS_KEYSTORE_LOCATION
oracle.wsm.security.util.SecurityConstants.ClientConstants.WSS_KEYSTORE_PASSWORD
```

5.2.5.2 SOAP and RESTful OWSM Security Policies

The options available when you select **OWSM Security Policies** differ slightly, depending on whether you are testing a SOAP web service or a RESTful web service.

The options for a SOAP web service are shown in Figure 5–11. The options for a RESTful web service are shown in Figure 5–12, with the **Advanced Options** field selected to display all of the available fields.

Figure 5-11 OWSM Security Policies Test Options for SOAP Web Services

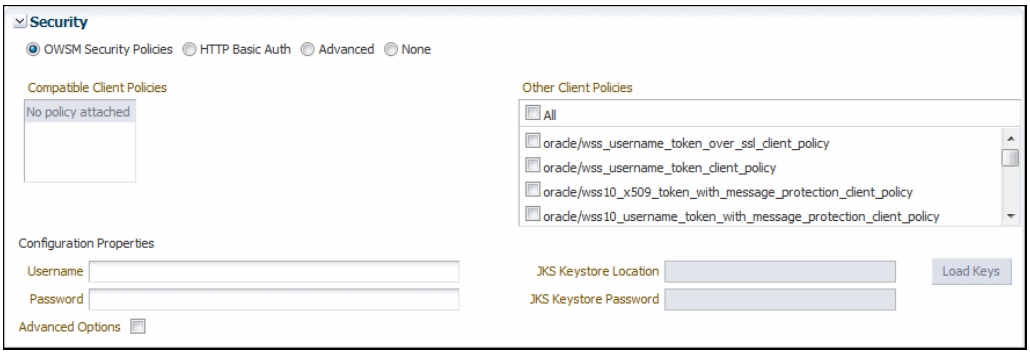
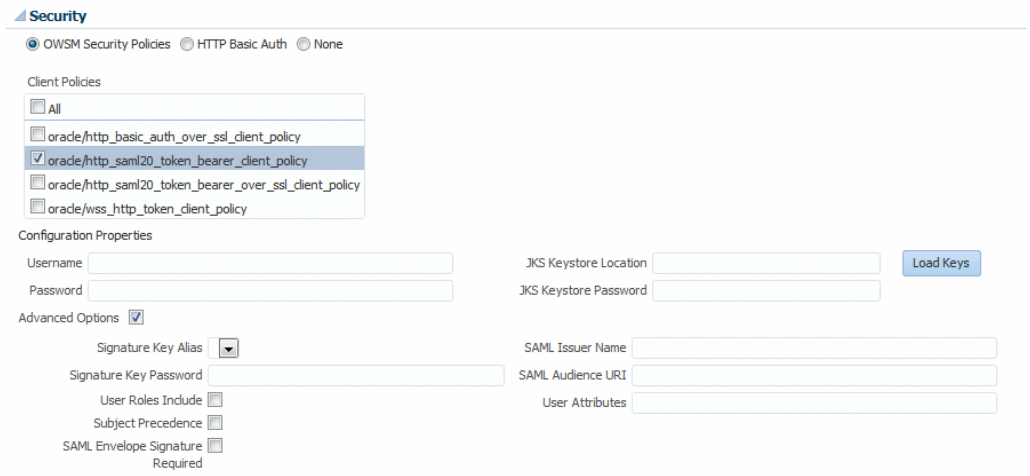


Figure 5-12 OWSM Security Policies Test Options for RESTful Web Services



To test the web service security using OWSM security policies:

1. Select the policies with which you want to test the service. The policies available differ, depending on whether you are testing a SOAP web service or a RESTful web service.

- **Selecting Policies for Testing a SOAP Web Service**

  From the **Compatible Client Policies** list, which displays the compatible client policies as specified in the WSDL, select the client policy to test.

  Alternatively, to perform a negative test on the endpoint, you can select a non-compatible policy, or **All** from the **Other Client Policies** list.

- **Selecting Policies for Testing a RESTful Web Service**

  From the **Client Policies** list, select the client policy to test.

The Configuration Properties required by the selected policy are indicated with an asterisk (*). For example:

- For username_token and http_token policies, the **Username** and **Password** fields are required; for SAML policies only the **Username** field is required.
- For message protection and SSL policies, the **JKS Keystore Location** and **JKS Keystore Password** fields are required. (The KSS keystore is not supported.)

2. Provide values for the required fields as determined by the policy.

   If the **JKS Keystore Location** and **JKS Keystore Password** fields are required by the policy, enter the location and password of a user-created keystore that is NFS accessible and click **Load Keys**. The associated keystore fields under **Advanced Options** are populated with the aliases specified in the keystore.

   For more information about creating a keystore, see "Generating Private Keys and Creating the Java Keystore" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

3. Click **Advanced Options**. Additional keystore alias and SAML properties fields are displayed. Properties required by the selected policies are indicated with an asterisk. Enter the required values, or provide override values in the applicable fields.

5.2.5.3 HTTP Basic Auth

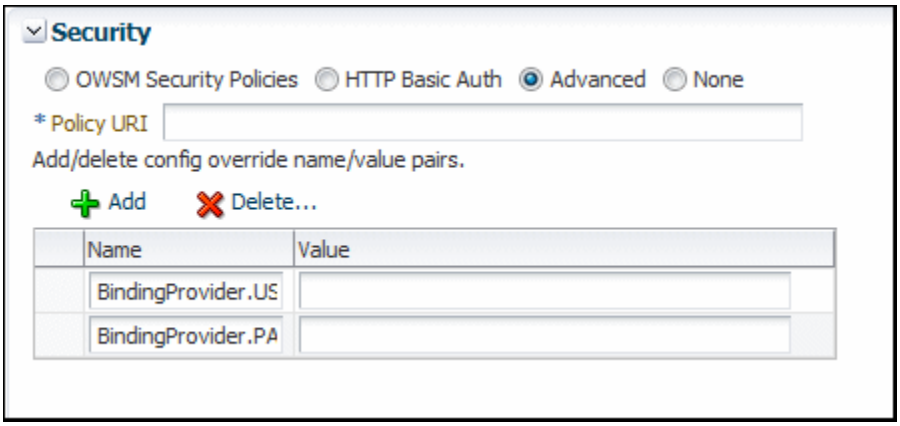
This option requires **Username** and **Password** credentials that are inserted into the HTTP transport header. The username and password must exist and be valid for the WebLogic Server.

5.2.5.4 Advanced

> **Note:**
>
> The Advanced option is not available for RESTful web services.

The options available when you select the **Advanced** option are shown in Figure 5–13.

Figure 5-13 Advanced Test Options



The Advanced option allows you to use a custom policy, including a copy of a predefined policy, to test the web service security. To do so:

1. Specify the URI of the custom policy in the **Policy URI** field. For example, oracle/wss11_message_protection_client_policy_copy. This field is required.
2. Specify any configuration overrides for the policy in the **Name** and **Value** fields.

   For username_token and http_token policies, Username and Password properties are required; for SAML policies only the username property is required.

   For message protection and SSL policies, the JKS Keystore Location and JKS Keystore Password properties are required. (The KSS keystore is not supported.)

   To add properties, click **Add** and provide the name/value pair for the configuration override. To delete a property, select it in the table and click **Delete**.

> **Note:**
>
> Properties must be specified using the full name of each property, for example `oracle.wsm.security.util.SecurityConstants.ClientConstants.WSS_KEYSTORE_LOCATION`. For a complete list of property names for properties that can be overridden, see "Overriding Client Policy Configuration Properties at Design Time" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
>
> All of the other policy attributes use the policy defaults. For example, if you enabled Secure Conversation or changed the message security algorithm suite, those policy values are used.

5.2.5.5 Supported Client Security Policies for SOAP Services

The following OWSM client security policies SOAP services are supported by the test function. Copies of these predefined policies are not directly listed and are available only through the Advanced option.

- `oracle/wss_http_token_client_policy`

- oracle/wss_http_token_over_ssl_client_policy
- oracle/wss_saml_token_bearer_over_ssl_client_policy
- oracle/wss_saml_token_over_ssl_client_policy
- oracle/wss_saml20_token_bearer_over_ssl_client_policy
- oracle/wss_saml20_token_over_ssl_client_policy
- oracle/wss_username_token_client_policy
- oracle/wss_username_token_over_ssl_client_policy
- oracle/wss10_message_protection_client_policy
- oracle/wss10_saml_token_with_message_integrity_client_policy
- oracle/wss10_saml_token_with_message_protection_client_policy
- oracle/wss10_saml_token_with_message_protection_ski_basic256_client_policy
- oracle/wss10_saml20_token_with_message_protection_client_policy
- oracle/wss10_username_id_propagation_with_msg_protection_client_policy
- oracle/wss10_username_token_with_message_protection_client_policy
- oracle/wss10_username_token_with_message_protection_ski_basic256_client_policy
- oracle/wss10_x509_token_with_message_protection_client_policy
- oracle/wss11_message_protection_client_policy
- oracle/wss11_saml_token_identity_switch_with_message_protection_client_policy
- oracle/wss11_saml_token_with_message_protection_client_policy
- oracle/wss11_saml20_token_with_message_protection_client_policy
- oracle/wss11_username_token_with_message_protection_client_policy
- oracle/wss11_x509_token_with_message_protection_client_policy

5.2.5.6 Supported Client Security Policies for RESTful Services

The following OWSM client security policies for RESTful services are supported by the test function:

- oracle/http_basic_auth_over_ssl_client_policy
- oracle/http_saml20_token_bearer_client_policy
- oracle/http_saml20_token_bearer_over_ssl_client_policy
- oracle/wss_http_token_client_policy
- oracle/http_jwt_token_client_policy
- oracle/http_jwt_token_over_ssl_client_policy

5.2.6 Enabling Quality of Service Testing

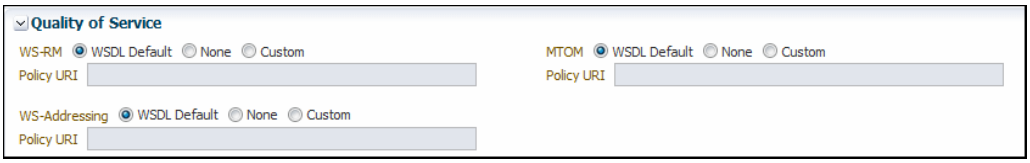
Three characteristics of Quality of Service (QoS) can be tested: Web services reliable messaging (WS-RM), WS-Addressing, and MTOM in the Quality of Service section of the Test Web Service Page.

> **Note:**
>
> This section is not applicable when testing RESTful web services.

For each type of Quality of Service, there are three options:

- **WSDL Default** – Execute the default behavior of the WSDL. For example, if **WSDL Default** is selected for MTOM, and the WSDL contains a reference to an MTOM policy, the policy is enforced. If the WSDL does not contain a reference to an MTOM policy, then no MTOM policy is enforced.
- **None** – No policy for the specific QoS, even if it is included in the WSDL, is executed. For example, if **None** is selected for WS-RM, no reliable messaging policy is enforced. If the WSDL contains a reference to a reliable messaging policy, it is ignored.
- **Custom** – Enforce a custom policy. For example, if a WS-Addressing policy is referenced in the WSDL, this policy will be ignored, and the policy specified in **URI** will be used instead.
- **URI** – Specify the location of the policy to be enforced.

Figure 5-14 Quality of Service Parameters on the Test Web Service Page



5.2.7 Testing HTTP Headers

The HTTP Header section allows you to add, modify, or delete HTTP headers to pass request information to a SOAP or RESTful web service.

After the service invocation, the HTTP headers should be displayed as part of the message response.

Figure 5-15 HTTP Header on the Test Web Service Page

5.2.8 Editing the Input Arguments as XML Source

You can view the input arguments in a user-friendly form, or you can edit the XML source code directly. If you edit the XML source directly, you must enter valid XML.

Use the drop-down list in the Input Arguments section of the page to toggle between **Tree View** and **XML View.**

Figure 5-16 Input Arguments - XML View

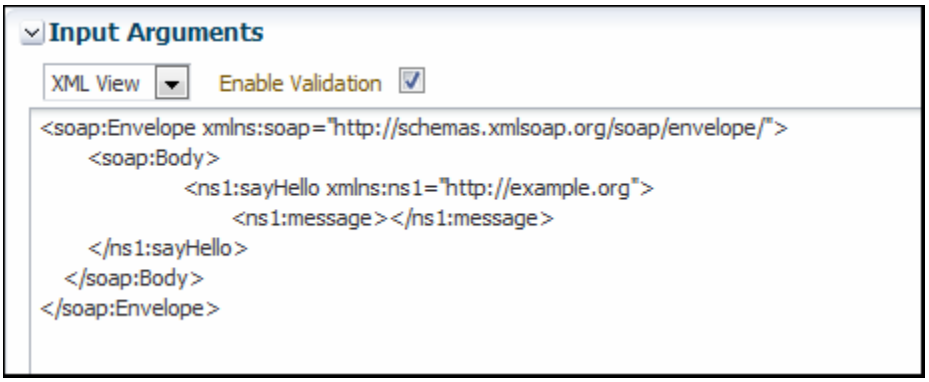
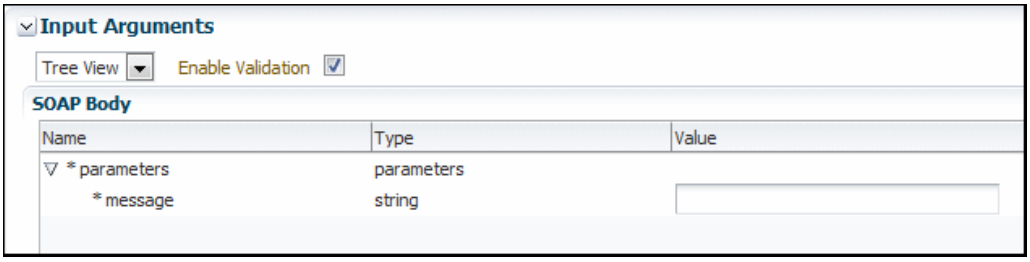


Figure 5-17 Input Arguments - Tree View

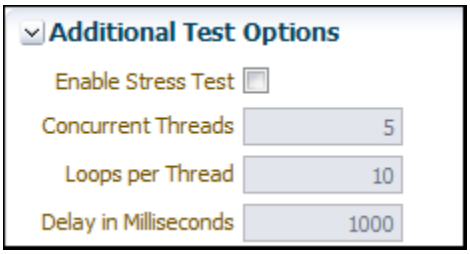


5.2.9 Stress Testing the Web Service Operation

Select the **Enable Stress Test** check box (Figure 5–18) to invoke a continuous series of invocations of the web service operation (Figure 5–18). The following options are available:

- **Concurrent Threads** – The number of concurrent threads on which the invocations should be sent. The default is 5 threads.
- **Loops per Thread** – The number of times to invoke the operation. The default is 10 times.
- **Delay in Milliseconds** – The number of milliseconds to wait between operation invocations. The default is 1000 milliseconds (1 second).

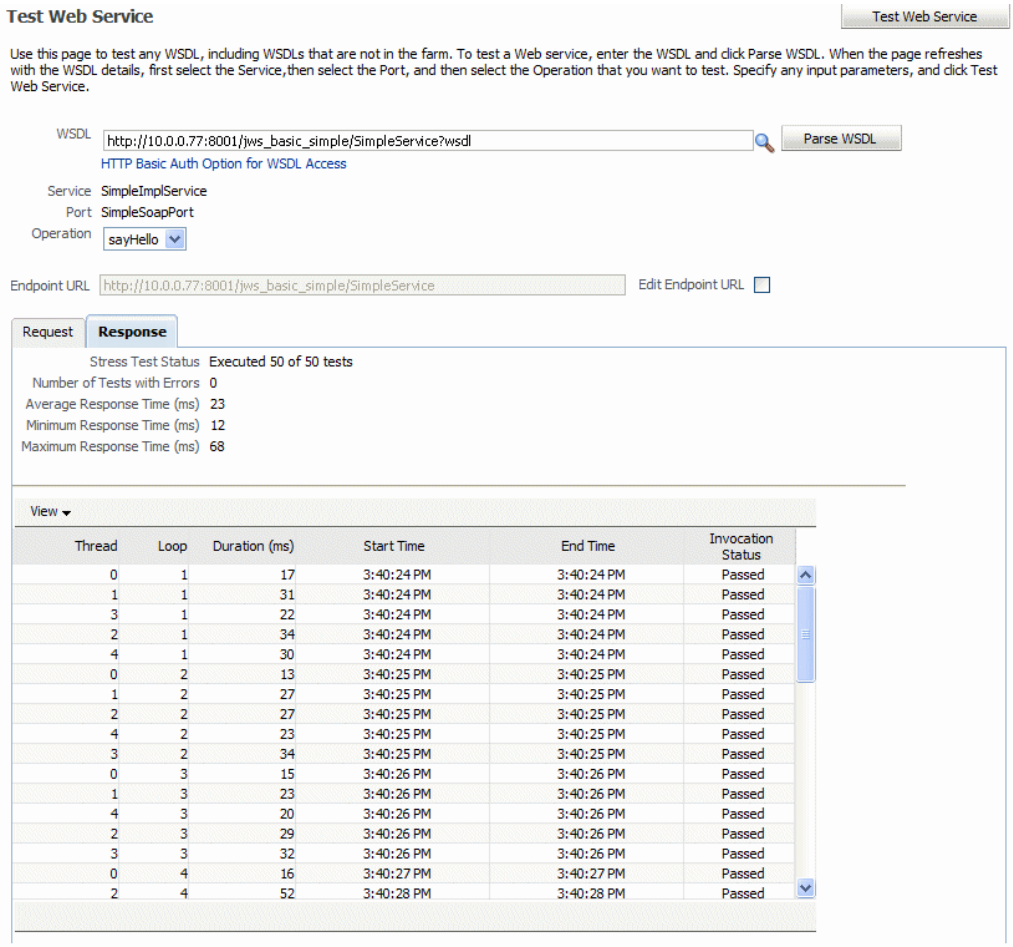
Figure 5-18 Stress Testing Parameters on the Test Web Service Page



When you invoke the test, a progress box indicates the test status. When the stress test is complete, a confirmation page displays the results of the test.

The **Response** tab provides additional information about the stress test, including the number of tests with errors, and the average, minimum, and maximum response times. Details about each test are provided in tabular form. For each test, you can view the thread and loop numbers, the duration of the test, the start and end times for the test, and the invocation status. You can filter the fields displayed in the table using the **View** menu.

Figure 5-19 Stress Test Results on Test Web Service Page



6
Monitoring and Auditing Web Services

From the Web Services application summary page in the Fusion Middleware Control, you can monitor web service faults and security failures, and configure web service ports.

> **Note:**
>
> Only a subset of the monitoring features described in this chapter apply to Java EE web services.

From the Web Services application summary page in Fusion Middleware Control, you can do the following:

- Monitor web services faults, including Security, Reliable Messaging, MTOM, Management, and Service faults.
- Monitor Security failures, including authentication, authorization, message integrity, and message confidentiality failures.
- Configure your web services ports, including enabling and disabling the port, attaching policies to web services, and enabling or disabling policies.

The Application home page also displays select web service details if the application includes web services.

For more information, refer to the following sections:

- Overview of Monitoring Web Services
- Auditing Web Services

6.1 Overview of Monitoring Web Services

This section contains the following sections:

- When Are Web Service Statistics Started or Reset?
- Viewing Web Service Statistics for a Server Instance
- Overview of Web Service Statistics for an Application
- Viewing Web Service Statistics for an Individual Web Service
- Viewing Operation Statistics for a Web Service Endpoint
- Viewing Statistics for a Java EE Web Service Operation
- Viewing Statistics for Java EE Web Service Clients
- Viewing Statistics for RESTful Resources
- Viewing Statistics for SOA Binding Components

- Overview of Viewing the Security Violations for a Web Service

In addition to the monitoring features described in this chapter, see "Analyzing Policy Usage" in *Securing Web Services and Managing Policies with Oracle Web Services Manager* to analyze how policies are used by one or more web services.

6.1.1 When Are Web Service Statistics Started or Reset?

The statistics described in this chapter are started or reset when any one of the following events occur:

- When the application is being deployed for the first time.
- When the application is redeployed.
- If the application is already deployed, and the hosting server is restarted.

6.1.2 Viewing Web Service Statistics for a Server Instance

The server-side web services page displays statistics for all of the web services on that server.

To view the web service statistics for a server:

1. In the navigation pane, expand **WebLogic Domain** to show the domain for which you want to see the policies and select the domain.
2. Expand the domain to show the servers in that domain. Select the server for which you want to view the statistics.
3. In the content pane, select **WebLogic Server**, and then **Web Services**.
4. The Web Services Server Summary page, which displays the statistics for the web services deployed on the server, is displayed.

   Depending on the types of web services you have deployed, tabs are available for the following web service types: Java EE, Oracle Infrastructure Web Services, and RESTful Services.

For Java EE web services, the following statistics are displayed in tabular format for each web service running on the server:

- Web Service Name—Name of the web service.
- Application name—Name of the application that contains the web service.
- Endpoint name—Name of the web service endpoint. Click the endpoint name to view the Web Service Endpoint page.
- Invocation count—Number of invocation requests to this endpoint.
- Response count—Number of responses generated from
- Response error count—Number of errors encountered during responses.
- Average execution time—Average time, in milliseconds, to execute the web service.
- Average response time—Average time, in milliseconds, to receive a response from the web service.

For Oracle Infrastructure web services, the following statistics are displayed in tabular format for each web service running on the server:

- Web Service Name—Name of the web service.
- Application name—Name of the application that contains the web service.
- Endpoint name—Name of the web service endpoint. Click the endpoint name to view the Web Service Endpoint page.
- Invocations completed—Total number of completed requests to this endpoint.
- Average invocation time—Average time (in milliseconds) for the web service to send a response, in milliseconds.
- Total faults—Total number of failed requests.

For RESTful services, the following statistics are displayed in tabular format for each web service running on the server:

- Application name—Name of the application that contains the RESTful web service.
- Module name—Name of the module for the RESTful service.
- Name—RESTful application name. Click the name to view the RESTful Service Application page.
- Invocation count—Number of times that the RESTful web service was invoked.
- Error count—Number of errors that the RESTful web service incurred.
- Average execution time—Average time (in milliseconds) for all RESTful web service executions.

6.1.3 Overview of Web Service Statistics for an Application

The following sections describe how to view web services statistics based on the type of application:

- "Viewing Web Service Statistics for a SOA Composite Application"
- "Viewing Web Service Statistics for a Non-SOA Oracle Infrastructure Web Service Application"
- "Viewing the Web Service Statistics for a Java EE Application"

6.1.3.1 Viewing Web Service Statistics for a SOA Composite Application

In Fusion Middleware Control, the dashboard for a SOA composite application displays the basic monitoring information for all services and references in the composite application, as shown in Figure 6-1.

To navigate to the dashboard page for a SOA composite application:

1. In the navigation pane, expand the **SOA** folder.
2. Expand **soa-infra** to view the SOA partitions, then expand the SOA partition (for example, the default partition) and select the target SOA composite application.

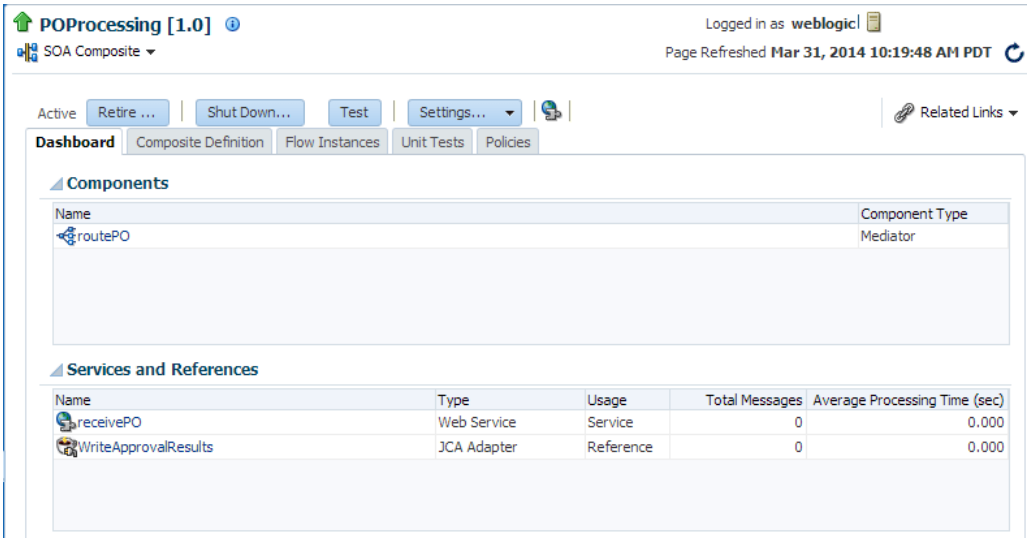
   The SOA composite home page displays.
3. Select the **Dashboard** tab if it is not already selected.

   The Components section of this tab lists the SOA components being used in the composite application, and the Services and References section displays the web service and reference bindings.

For the SOA composite services and references, the following web service application-level statistics are displayed:

- Name—Name of the service or reference.
- Type—Type of service or reference
- Usage—Service or reference.
- Total Messages—Total number of messages.
- Average Processing Time (sec)—Average processing time, in seconds.

Figure 6-1 Dashboard for SOA Composite Application



6.1.3.2 Viewing Web Service Statistics for a Non-SOA Oracle Infrastructure Web Service Application

In Fusion Middleware Control, the Web Services summary page for an application displays the collective **Summary** and fault/violation information for all web services in the application, as shown in Figure 6-2.

The **Charts** section shows a graphical view of all security faults for a web service.

To navigate to the Web Service Summary page for a non-SOA Oracle Infrastructure web service application:

1. In the navigation pane, expand the **Application Deployments** folder to expose the applications in the domain and select the application deployment.

   The Domain Application Deployment home page is displayed in the content pane.

2. In the navigation pane, expand the application deployment and select the application name.

   The Application Deployment home page is displayed in the content pane.

3. In the content pane, select **Application Deployment**, then **Web Services**.

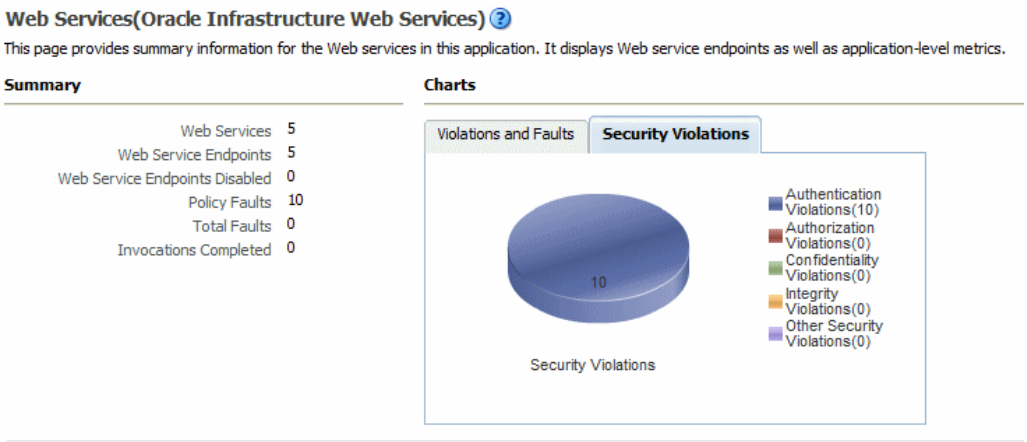
   The Web Services Summary page for an application is displayed.

The page displays web service endpoints as well as application-level metrics.

For Oracle Infrastructure web services, the following web service application-level statistics are displayed:

- Web Services—Total number of web services in the application.
- Web Service Endpoints—Total number of endpoints used by web services in this application.
- Web Service Endpoints Disabled—Total number of endpoints assigned to web services which have been disabled.
- Policy Faults—Number of web service requests that failed due to a policy fault. Specifies the total number since the application was last restarted.
- Total Faults—Total number of failed requests, including security, reliable messaging, MTOM, management, and service faults. Specifies the total number since the application was last restarted.
- Invocations Completed—Total number of client requests to the web service since the application was last restarted.

Figure 6-2 Web Services Performance Summary and Charts for an Application



6.1.3.3 Viewing the Web Service Statistics for a Java EE Application

In Fusion Middleware Control, the Web Services summary page for a Java EE application, including SOAP and RESTful services, displays the collective Summary and fault/violation information for all web services in the application, as shown in Figure 6-3.

To navigate to the Web Service Summary page for a Java EE web service application:

1. In the navigation pane, expand the **Application Deployments** folder to expose the applications in the domain and select the application deployment.

   The Domain Application Deployment home page is displayed in the content pane.

2. In the navigation pane, expand the application deployment and select the application name.

   The Application Deployment home page is displayed in the content pane.

3. In the content pane, select **Application Deployment**, then **Web Services**.

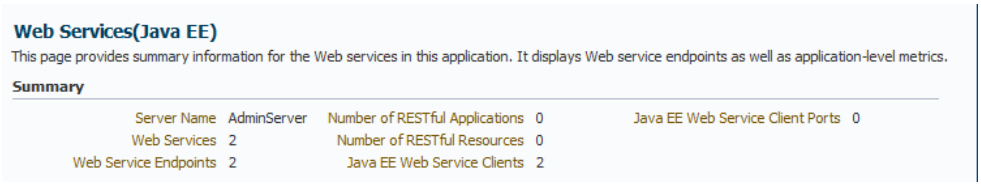
   The Web Services Summary page for an application is displayed.

   The page displays web service endpoints as well as application-level metrics.

For Java EE web services, including SOAP and RESTful web services, the following web service application-level statistics are displayed:

- Server Name—Server on which the application is deployed.
- Web Services—Number of web services in the application.
- Web Service Endpoints—Total number of endpoints used by web services in this application.
- Number of RESTful Applications—Total number of RESTful applications registered with this web service.
- Number of RESTful Resources—The number of resources available to the RESTful application.
- Java EE Web Service Clients—Number of run-time client instances in the application.
- Java EE Web Service Client Ports—Number of web service client ports in the application to which you can attach OWSM policies.

Figure 6-3 Java EE Web Services Summary



6.1.4 Viewing Web Service Statistics for an Individual Web Service

The **Web Service Details** section of the Web Services Summary page for an application displays statistics on a per-web service basis, as shown in Figure 6-4. For information about navigating to the Web Services Summary page for an application, see "Overview of Web Service Statistics for an Application".

The following statistics are displayed for Java EE web services:

- Name—Name of the web service. Expand the web service to display the web service endpoint.
- Invocation Count—Number of invocation requests to this endpoint.
- Response Error Count—Number of errors encountered during responses.
- Response Count—Number of responses
- Average Execution Time (ms)—Average time, in milliseconds, to execute the web service.
- Average Response Time (ms)—Average time, in milliseconds, to receive a response from the web service.

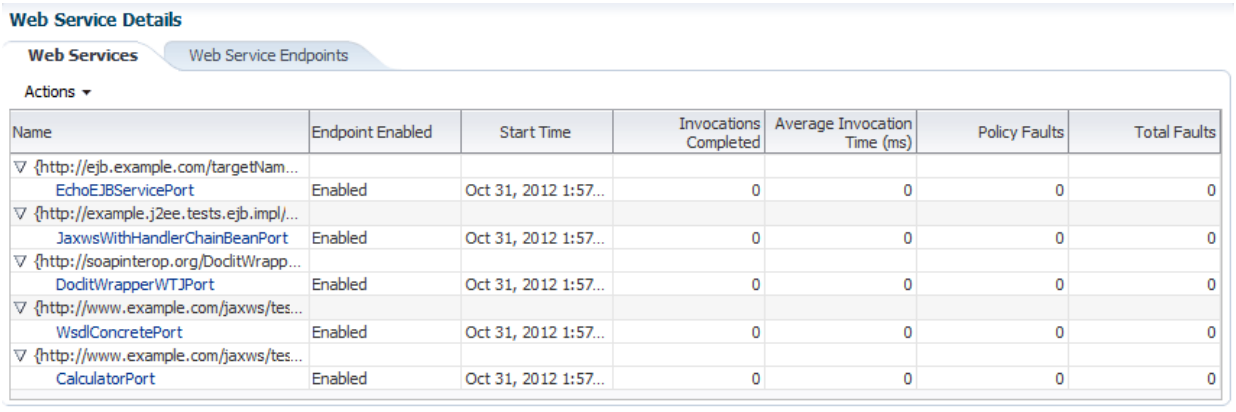
The following statistics are displayed for RESTful web services:

- Module Name and RESTful Application Name—Name of the module and RESTful application. Click the RESTful application name to view the RESTful Service Application page.
- Resource Name—Name of the RESTful resource.
- Resource Type—Type of the RESTful resource.
- Resource Path—URI of the RESTful resource.
- Invocation Count—Number of invocation requests to this endpoint.
- Average Execution Time (ms)—Average time, in milliseconds, to execute web services.

The following statistics are displayed for Oracle Infrastructure web services:

- Name—Name of the web service. Expand the web service to display the web service endpoint.
- Endpoint Enabled—Flag that specifies whether the web service is enabled or disabled. For Oracle Infrastructure web service providers, this field displays n/a.
- Start Time—Time the web service was started.
- Invocations Completed—Number of completed requests to this endpoint.
- Average Invocation Time—Average time for all web service invocations to be processed.
- Policy Faults—Number of failed requests because a policy was not successfully executed.
- Total Faults—Total number of failed requests.

Figure 6-4 Web Service Statistics for Individual Oracle Infrastructure Web Services



6.1.5 Viewing Operation Statistics for a Web Service Endpoint

Follow this procedure to view statistics for a web service endpoint. To view statistics for individual operations, see "Viewing Statistics for a Java EE Web Service Operation".

To display operation statistics for a particular web service endpoint:

1. Navigate to the Web Service Summary page as described in "Viewing the Web Services Summary Page for an Application".
2. In the Web Services Details section of the Web Services summary page, select the **Web Service Endpoints** tab.
3. Select the endpoint for which you want to display the statistics.

   The Web Service Endpoint page is displayed.
4. Select the **Operations** tab if it is not already selected.

   The following statistics are presented for Oracle Infrastructure web services:

Element	Description
Operation Name	Name of the operation.
One Way	Flag that specifies whether the operation returns a value to the calling operation.
Action	URI of the action.
Input Encoding	Encoding style of the input message.
Output Encoding	Encoding style of the output message.
Invocations Completed	Number of completed requests to this endpoint.
Average Invocation Time	Average time for all web service invocations to be processed.
Faults	Total number of faults for this endpoint.

The following statistics are presented for Java EE web services:

Element	Description
Name	Name of the operation.
Invocation Count	Number of times that the web service was invoked.
Average Dispatch Time (ms)	Average time, in milliseconds, for all web service invocations to be processed.
Average Execution Time (ms)	Average time, in milliseconds, for all web service executions.
Average Response Time (ms)	Average time, in milliseconds, for all responses generated.
Response Count	Total number of responses generated from the web service invocations.
Response Error Count	Total number of errors from responses generated from the web service invocations.

6.1.6 Viewing Statistics for a Java EE Web Service Operation

The individual web service operations are displayed on the **Operations** tab of the Web Service Endpoint page. This procedure applies only to Java EE web service operations.

To view the statistics for an individual Java EE web service operation:

1. Navigate to the Web Service Operation page as described in "Viewing the Details for a Java EE Web Service Operation".

2. Click the name of an operation to view its statistics.

The Web Service Operation page displays the following statistics:

Element	Description
Application Name	The name of the application that this operation is associated with.
Web Service Name	The name of the web service that this operation is associated with.
Endpoint Name	The name of the endpoint that this operation is associated with.
Operation Name	The name of the web service operation.
Endpoint URI	The URI of the endpoint that this operation is associated with.

Errors

The Errors section of the Web Service Operation page displays the following error statistics:

Element	Description
Error Count	Number of errors sending or receiving a request.
Last Error	Last error that occurred processing a request.
Last Error Time	Time on WebLogic Server of the last error for a request (sending or receiving) was detected expressed as the number of milliseconds since midnight, January 1, 1970 UTC.
Response Error Count	Total number of errors from responses generated from operation invocations.
Last Response Error	Last response error to arrive for this client/service (or null if no errors have occurred).
Last Response Error Time	Time on WebLogic Server of the last error sending or receiving a response (or 0 if no failures have occurred) expressed as the number of milliseconds since midnight, January 1, 1970 UTC.

Invocation Statistics

The Invocation Statistics section of the Web Service Operation page displays the following invocation statistics:

Element	Description
Invocation Count	Total number of operation invocations in the current measurement period.
Last Invocation Time	Time of the last operation request to be sent or received (or 0 if no requests have been sent or received).
Average Dispatch Time (ms)	Average operation dispatch time (in milliseconds) for the current measurement period. Dispatch time refers to the time for WebLogic Server to process the invocation. The measurement period typically starts when WebLogic Server is first started.
Dispatch Time Total (ms)	Total time (in milliseconds) for all operation dispatches in the current measurement period. Dispatch time refers to the time for WebLogic Server to process the invocation. The measurement period typically starts when WebLogic Server is first started.

Element	Description
Dispatch Time High	Longest operation dispatch time for the current measurement period. Dispatch time refers to the time for WebLogic Server to process the invocation. The measurement period typically starts when WebLogic Server is first started.
Dispatch Time Low	Shortest operation dispatch time for the current measurement period. Dispatch time refers to the time for WebLogic Server to process the invocation. The measurement period typically starts when WebLogic Server is first started.
Average Execution Time (ms)	Average operation execution time (in milliseconds).
Execution Time Total (ms)	Total time (in milliseconds) for all operation executions.
Execution Time High	Longest operation execution time.
Execution Time Low	Shortest operation execution time.

Response Statistics

The Response Statistics section of the Web Service Operation page displays the following response statistics:

Elements	Description
Response Count	Total number of responses generated from operation invocations.
Last Response Time	Time on WebLogic Server of the last response to arrive for this client/service (or 0 if no responses have been received) expressed as the number of milliseconds since midnight, January 1, 1970 UTC.
Average Response Time (ms)	Average response time (in milliseconds) from the responses generated from operation invocations.
Response Time Total (ms)	Total time (in milliseconds) for all responses generated from operation invocations.
Response Time High	Longest response time from the responses generated from operation invocations.
Response Time Low	Lowest response time from the responses generated from operation invocations.

6.1.7 Viewing Statistics for Java EE Web Service Clients

To display web service statistics for the run-time client instances in a Java EE application:

1. Navigate to the Java EE web service application summary page, as described in "Viewing the Web Services Summary Page for an Application".
2. Select the **Java EE Web Service Clients** tab to view the clients in the application.

> **Note:**
>
> This tab is available only if the application contains Java EE web service clients.

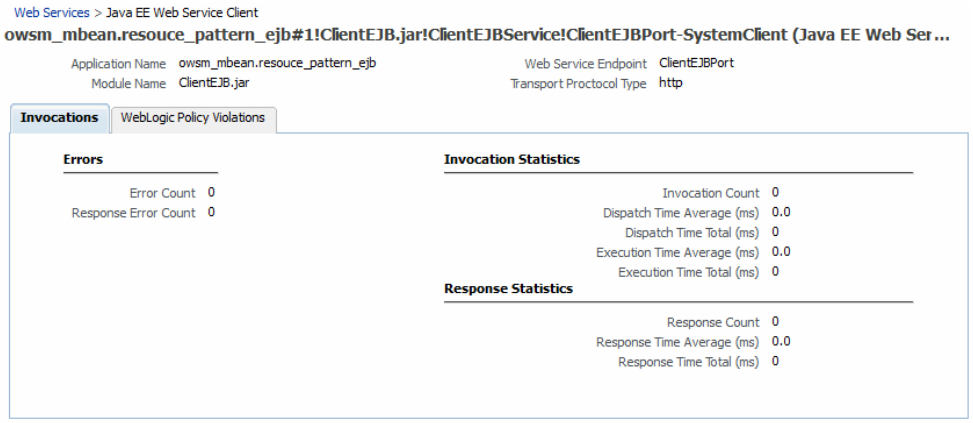
3. Select the **Monitoring** tab, if it is not already selected to view the statistics for all run-time client instances in the application.

> **Note:**
>
> For JAX-WS web services, the web services run time creates system-defined client instances within a web service endpoint that are used to send protocol-specific messages as required by that endpoint. These client instances are named after the web service endpoint that they serve with the following suffix: `-SystemClient`. Monitoring information relevant to the system-defined client instances is provided to assist in evaluating the application.

4. Select the client in the Client column to display web service statistics for that client.

   The Java EE Web Service Client page is displayed, as shown in Figure 6-5.

Figure 6-5 Java EE Web Service Client Statistics



   The following summary information is presented for the run-time client instance.

   - Application Name—The name of the application with which the client is associated.
   - Module Name—Name of the Java EE module in which the endpoint is running.
   - Web Service Endpoint—Name of the port which the client invokes.
   - Transport Protocol Type—Transport protocol required by the service.

5. Select the **Invocations** tab to view the invocation statistics for the client.

   Table 6-1 lists the invocation statistics displayed for the run-time client instance.

Table 6-1 Invocation Statistics for Java EE Web Service Client

Element	Description
Errors	

Table 6-1 (Cont.) Invocation Statistics for Java EE Web Service Client

Element	Description
Error Count	Total number of security faults and violations.
Response Error Count	Total number of errors from responses generated from invocations of this client instance
Invocation Statistics	
Invocation Count	Total number of times that operations on service side have been invoked by the client instance in the current measurement period.
Average Dispatch Time (ms)	Average dispatch time for the current measurement period.
Dispatch Time Total (ms)	Total time for all dispatches of this operation in the current measurement period.
Average Execution Time (ms)	Average execution time of this operation.
Execution Time Total (ms)	Total time for all executions of this operation
Response Statistics	
Response Count	Total number of responses generated from invocations of this operation.
Average Response Time (ms)	Average response time from the responses generated from invocations of this operation.
Response Time Total (ms)	Total time for all responses generated from invocations of this operation.

6. Select the **WebLogic Policy Violations** tab to view the policy violations for this client run-time instance.

> **Note:**
>
> This tab appears only if there are WebLogic web service policies attached to the Java EE web service client.

Table 6-2 lists the policy violations for the client run-time instance.

Table 6-2 WebLogic Policy Violations for Java EE Web Service Client

Element	Description
Summary	
Total Faults	Total number of failed requests.
Policy Faults	Total number of policy faults.
Total Security Faults	Total number of security faults and violations.
Violations	
Authentication Violations	Total number of authentication violations generated for this port. Only incoming message processing can add to the violation count.

Table 6-2 (Cont.) WebLogic Policy Violations for Java EE Web Service Client

Element	Description
Confidentiality Violations	Total number of confidentiality violations generated for this port. Both outgoing and incoming message processing can add to the violation count.
Integrity Violations	Total number of integrity violations generated for this port. Both outgoing and incoming message processing can add to the violation count.
Successes	
Authentication Successes	Total number of authentication successes detected for this port. Only incoming message processing can add to the success count.
Confidentiality Successes	Total number of confidentiality successes generated for this port. Both outgoing and incoming message processing can add to the success count.
Integrity Successes	Total number of integrity successes generated for this port. Both outgoing and incoming message processing can add to the success count.

6.1.8 Viewing Statistics for RESTful Resources

To display web service statistics for the resources in a RESTful web service:

1. Navigate to the Web Services application summary page, as described in "Viewing the Web Services Summary Page for an Application".
2. Select the **RESTful Services** tab to view the RESTful applications.

> **Note:**
>
> This tab is available only if the application contains RESTful web services.

3. Click the RESTful application name for which you want to view RESTful resources.
4. In the RESTful Resources tab, click the resource for which you want to view statistics.

   Table 6-3 lists the summary information that is provided.

Table 6-3 Summary of RESTful Resource

Field	Description
Application Name	The name of the application with which the RESTful service is associated.
Module Name	Name of the module in which the RESTful application is running.
RESTful Application Name	Name of the RESTful application.
Resource Name	URI of the RESTful resource.

Table 6-3 (Cont.) Summary of RESTful Resource

Field	Description
Resource Type	Type of the resource.
Resource Path	Path of the resource.
Number of Methods	Number of methods.
Number of Subresource locators	Number of subresource locators.
Invocation Count	Number of invocations of the RESTful service.
Average Execution Time (ms)	Average execution time, in milliseconds, of this method.

5. In the **RESTful Methods** tab, view the statistics for the methods.

   Table 6-4 lists the statistics for each RESTful method.

Table 6-4 Summary Statistics for RESTful Resources

Field	Description
Method Name	Name of the method.
Method & Request Statistics	Eye-glass icon to drill down for method and request data.
Return Type	Return type of the method.
Path	Path of the method.
HTTP Method	HTTP method to which the method is mapped.
Producing Media Type	Total time for all dispatches of this operation in the current measurement period.
Invocation Count	Number of invocations of the RESTful method.
Average Execution Time (ms)	Average execution time, in milliseconds, of this operation.
Execution Time Total (ms)	Total time for all executions of the method.

6. Still in the **RESTful Methods** tab, click the eye-glass icon in the **Method & Request Statistics** column to view the following request and method statistics for a specific RESTful method:

Table 6-5 Method and Request Statistics for RESTful Resources

Field	Description
Average Request Processing Time (ms)	Average request processing time in milliseconds.
Maximum Request Processing Time (ms)	Maximum time to process request in milliseconds.
Minimum Request Processing Time (ms)	Minimum time to process request in milliseconds.
Total Request Count	Total number of requests that have been processed.

Table 6-5 (Cont.) Method and Request Statistics for RESTful Resources

Field	Description
Total Request Rate (per ms)	Total time needed to process requests, per millisecond.

7. When you are finished viewing the statistical information, click **OK**.

6.1.9 Viewing Statistics for SOA Binding Components

You can monitor service and reference binding components in SOA composite applications, including instances, faults, and rejected messages. For complete details, refer to the following sections in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*:

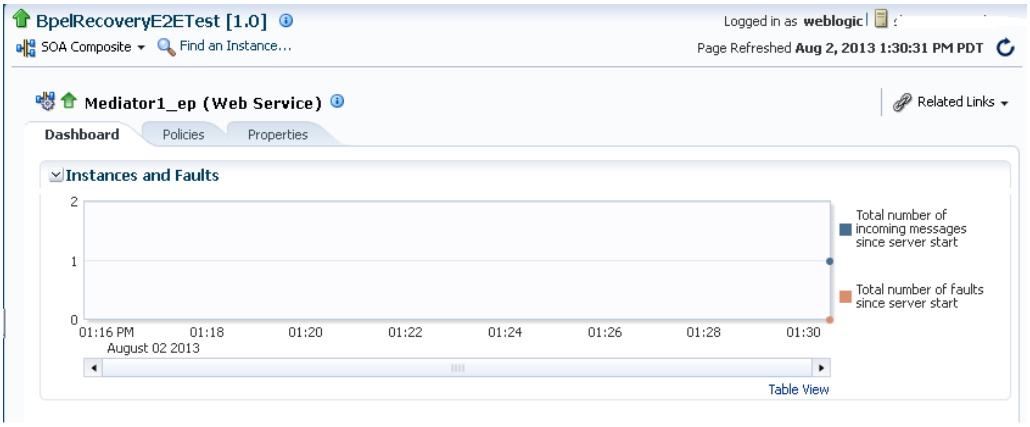
- "Monitoring Service and Reference Binding Components"
- "Monitoring Service and Reference Binding Components in the SOA Infrastructure"

To view statistics for SOA binding components:

1. Navigate to the SOA composite application, as described in "Viewing the Web Services and References in a SOA Composite".
2. In the Services and References section, select a specific service or reference.

If you select a service binding component that is a JCA adapter, web service, or REST service, the Dashboard page displays a graphic representation of the total number of incoming messages and faults since server startup, as shown in Figure 6-6.

Figure 6-6 Statistics for SOA Binding Components



6.1.10 Overview of Viewing the Security Violations for a Web Service

Follow the procedures listed below to view security violations for a web service:

- Viewing the Security Violations for an Oracle Infrastructure Web Service
- Viewing the Security Violations for a Java EE JAX-WS Web Service

- Viewing the Security Violations for a Java EE JAX-RPC Web Service

6.1.10.1 Viewing the Security Violations for an Oracle Infrastructure Web Service

To view the security violations for an Oracle Infrastructure web service:

1. Navigate to the Web Services Summary page as described in "Viewing the Web Services Summary Page for an Application".
2. In the Charts section of the page, select the **Security Violations** tab.

   A graphical representation of the authentication, authorization, confidentiality, and integrity faults for all web services in the application is displayed in the pie chart.
3. In the Web Service Details section of the page, expand the web service to display the web service endpoints if they are not already displayed.
4. Click the name of the endpoint to navigate to the Web Service Endpoint page.
5. Click the **Charts** tab to see a graphical representation of all faults and all security violations for the endpoint.
6. Click the **OWSM Policies** tab.

   Two tables are displayed.

   The Globally Attached Policies table displays the name of the policy and the policy set that references it.

   The Directly Attached Policies table displays the name of the policy and the policy status (whether the policy is enabled or disabled).

   Both tables list the category to which the policy belongs (security, MTOM attachments, reliable messaging, WS-addressing, and management).

   Table 6-6 lists the violation information provided for each type of policy attachment.

Table 6-6 Policy Violation Information for an Endpoint

Violation Type	Description
Total Violations	Total number of faults for this policy. **Note**: Total violations may not be equal to the sum of the security violations shown below (for example, Authentication, Authorization, Confidentiality, and Integrity). Other security violations that do not fall into these major categories and non-security violations are also captured in the total violations count.
Security Violations	
Authentication	Number of authentication failures since the server was restarted.
Authorization	Number of authorization failures since the server was restarted.
Confidentiality	Number of message confidentiality failures since the server was restarted.
Integrity	Number of message integrity failures since the server was restarted.

6.1.10.2 Viewing the Security Violations for a Java EE JAX-WS Web Service

To view the security violations for a Java EE JAX-WS web service:

1. Navigate to the Web Services Summary page as described in "Viewing the Web Services Summary Page for an Application".
2. In the Web Service Details section of the page, expand the web service to display the web service endpoints if they are not already displayed.
3. Click the name of the endpoint to navigate to the Web Service Endpoint page.
4. Do one of the following, depending on the type of policies attached to the endpoint:
   - If OWSM policies are attached to the endpoint, click the **OWSM Policies** tab.

     A list of the policies that are attached to the endpoint is displayed. For each policy, the table displays the name of the policy, the category of the policy (security, MTOM attachments, reliable messaging, WS-addressing, and management), and the policy status (whether the policy is enabled or disabled). Table 6-6 describes the violation information that is displayed for each OWSM policy attached to the endpoint.
   - If WebLogic policies are attached to the endpoint, click the **WebLogic Policy Violations** tab.

     This tab shows policy violation details about WebLogic policies attached to a JAX-WS endpoint. Table 6-7 describes the information provided on this page.

Table 6-7 WebLogic Policy Violation Data

Element	Description
Summary	
Total Faults	Total number of failed requests.
Policy Faults	Number of failed requests because a policy was not successfully executed.
Total Violations	Total number of faults for this policy.
Violations	
Authentication Violations	Number of authentication failures since the server was restarted.
Confidentiality Violations	Number of message confidentiality failures since the server was restarted.
Integrity Violations	Number of message integrity failures since the server was restarted.
Successes	
Authentication Successes	Number of authentication successes since the server was restarted.
Confidentiality Successes	Number of message confidentiality successes since the server was restarted.
Integrity Successes	Number of message integrity successes since the server was restarted.

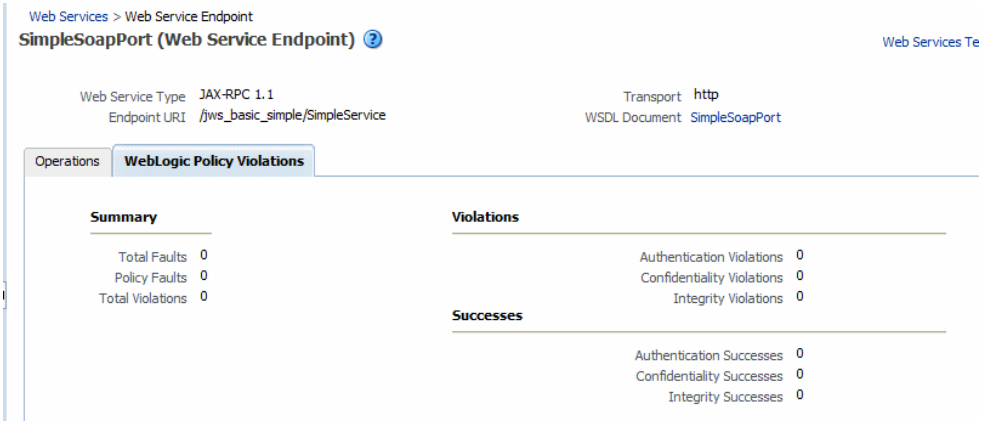
6.1.10.3 Viewing the Security Violations for a Java EE JAX-RPC Web Service

To view the security violations for a Java EE JAX-RPC web service:

1. Navigate to the Web Services Summary page for the application.
2. In the Web Service Details section of the page, expand the web service to display the web service endpoints if they are not already displayed.
3. Click the name of the endpoint to navigate to the Web Service Endpoint page.
4. Click the **WebLogic Policy Violations** tab.

   This tab shows policy violation details about WebLogic policies attached to a JAX-RPC endpoint, as shown in Figure 6-7. For a description of the information displayed on this tab, see Table 6-7.

Figure 6-7 Security Violations for a Java EE JAX-RPC Web Service Endpoint



6.2 Auditing Web Services

Auditing describes the process of collecting and storing information about security events and the outcome of those events. An audit provides an electronic trail of selected system activity.

An audit *policy* defines the type and scope of events to be captured at run time. Although a very large array of system and user events can occur during an operation, the events that are actually audited depend on the audit policies in effect at run time. You can define component- or application-specific policies, or audit individual users.

You configure auditing for system components, including web services, and applications at the domain level using the Audit Policy page. You can audit SOA and ADF services.

The following table summarizes the events that you can audit for web services and the relevant component.

Table 6-8 Auditing Events for Web Services

Enable auditing for the following web service events. . .	Using this system component. . .
• User authentication. • User authorization. • Policy enforcement, including message confidentiality, message integrity, and security policy.	OWSM—Agent For more information, see "OWSM-AGENT Events and Attributes".
• Web service requests sent and responses received. • SOAP faults incurred. **Note**: In this case, events are logged for both security and non-security web service invocations.	Oracle web services For more information, see "Oracle Web Services Events and Attributes".
• OWSM assertion template creation, deletion, or modification. • OWSM policy intent creation, deletion, or modification. • OWSM policy creation, deletion, or modification. • OWSM policy set authoring creation, deletion, or modification.	OWSM—Policy Manager **Note:** The Policy Manager audits both local policy attachments and global policy attachments for policy sets. For more information, see "OWSM-PM-EJB Events and Attributes".
• OWSM policy attachment.	OWSM—Policy Attachment **Note:** The Policy Attachment audits only local policy attachments. For more information, see "Web Services Policy Attachment Events and Attributes".

You can also audit the events for a specific user, for example, you can audit all events by an administrator.

For more information about configuring audit policies, see "Configuring and Managing Auditing" in *Securing Applications with Oracle Platform Security Services*.

The following sections describe how to define audit policies and view audit data:

- Configuring Audit Policies
- Managing Audit Data Collection and Storage
- Viewing Audit Reports

6.2.1 Configuring Audit Policies

Follow the steps in this section to configure audit policies. For more information, see "Manage Audit Policies for Java Components with Fusion Middleware Control" in *Securing Applications with Oracle Platform Security Services*.

1. From the WebLogic Domain menu, select **Security > Audit Policy**.

   The Audit Policy Settings page is displayed.

The audit policies table, at the center of the page, displays the audits that are currently in effect.

2. Select the component that you want to audit from the Audit Component Name menu.
3. Select an audit level from the Audit Level menu.

   Valid audit levels include:

   - None—Disables auditing.
   - Low, Medium, High—Audits subsets of event categories representing pre-defined levels of auditing.
   - Custom—Enables you to provide a custom auditing policy.

   You can view the components and applications that are selected for audit at each level in the audit policies list. For all audit levels other than Custom, the information in the audit policies list is greyed out, as you cannot customize other audit level settings.
4. To customize the audit policy, select the Custom option and perform one of the following steps:

   - Select the information that you want to audit by clicking the associated checkbox in the Select for Audit column.

     You can audit at the following levels of granularity: All events for a component, all events within a component event category, an individual event, or a specific outcome of an individual event (such as, success or failure).

     Click **Select All** to select all categories, **None** to deselect all categories, or **Audit All Events** to audit all events, including specific outcome of individual events (such as, successes and failures).

     At the event outcome level, you can specify an edit filter. Filters are rules-based expressions that you can define to control the events that are returned. For example, you might specify an Initiator as a filter for policy management operations to track when policies were created, modified, or deleted by a specific user. To define a filter for an outcome level, click the **Edit Filter** icon in the appropriate column, specify the filter attributes, and click **OK**. The filter definition appears in the Filter column.

     Deselect the checkbox for a component at a higher level to customize auditing for its subcomponents. You can select all components and applications by checking the checkbox adjacent to the column name.
   - At the event outcome level, you can specify an edit filter. Filters are rules-based expressions that you can define to control the events that are returned. For example, you might specify an Initiator as a filter for policy management operations to track when policies were created, modified, or deleted by a specific user. To define a filter for an outcome level, click the **Edit Filter** icon in the appropriate column, specify the filter attributes, and click **OK**. The filter definition appears in the Filter column.
   - To audit only success or failures for all system components and applications, select **Select Successes Only** or **Select Failures Only** from the Select menu, respectively. To clear all selections, select **None**.
5. If required, enter a comma-separated list of users in the Users to Always Audit text box.

Specified users will always be audited, regardless of whether auditing is enabled or disabled, and at what level auditing is set.

6. Click **Apply.**

   To revert all changes made during the current session, click **Revert**.

6.2.2 Managing Audit Data Collection and Storage

To manage the data collection and storage of audit information, you need to perform the following tasks:

- Set up and manage an audit data repository.

  You can store records using one of two repository modes: file and database. It is recommended that you use the database repository mode. The Oracle Business Intelligence Publisher-based audit reports only work in the database repository mode.

- Set up audit event collection.

For more information, see "Managing the Audit Data Store" in *Securing Applications with Oracle Platform Security Services*.

6.2.3 Viewing Audit Reports

For database repositories, data is exposed through pre-defined reports in Oracle Business Intelligence Publisher.

A number of predefined reports are available, such as: authentication and authorization history, OWSM policy enforcement and management, and so on. For details about generating and viewing audit reports using Oracle Business Intelligence Publisher, see "Using Audit Analysis and Reporting" in *Securing Applications with Oracle Platform Security Services*.

For file-based repositories, you can view the bus-stop files using a text editor and create your own custom queries.

7

Managing Diagnostic and Message Logs

This chapter describes how to manage and configure diagnostic and message logs. Oracle Fusion Middleware components, including web services, generate log files containing messages that record all types of events, including startup and shutdown information, errors, warning messages, access information on HTTP requests, and so on. Each log message includes specific information such as time, component ID, and user to assist you in pinpointing and diagnosing problems that arise.
This chapter includes the following sections:

7.1 Introduction to Diagnosing Problems Using Logs

You can review log messages to diagnose problems with specific components, such as web services.

There are two categories of log files that you can reference to assist in diagnosing problems with web services:

- **Diagnostic logs**—Enable you to access diagnostic data about specific feature components in Oracle Fusion Middleware. For more information, see "Overview of Diagnostic Logs for Web Services".

  There is a set of predefined diagnostic loggers. You can configure your own diagnostic logger, as described in "Configuring Log Files for a Web Service".

- **Message logs**—Enable you to view elements of the SOAP message request. You control message log creation using policies. For more information, see "Overview of Message Logs for Web Services".

For more information about logging in Oracle Fusion Middleware, see "Managing Log Files and Diagnostic Data" in *Administering Oracle Fusion Middleware*.

The following sections describe how to use diagnostic and message logs to diagnose problems. A set of sample logs is provided at the end of this section.

7.1.1 Overview of Diagnostic Logs for Web Services

Diagnostic logs enable you to access diagnostic data about specific feature components in Oracle Fusion Middleware.

The following sections describe how to view and manage diagnostic log files:

- Viewing Diagnostic Logs
- Filtering Diagnostic Logs
- Logging OWSM Debug Messages

7.1.1.1 Setting the Log Level for Diagnostic Logs

You set the logging level for web service and OWSM components at the WebLogic Server level, using the Log Configuration page.

In addition, you can override the log levels set at the server level for a specific web service endpoint from the Web Service Endpoint page. The logging level set at the web service endpoint level must be "finer grained" than the level set at the WebLogic Server level. Otherwise, the logging level set at the WebLogic Server level will be used.

The following procedures describe how to set the log level for diagnostic logs at the WebLogic Server and web service endpoint levels. For more information, see "Setting the Level of Information Written to Log Files" in *Administering Oracle Fusion Middleware*.

To set the log level for diagnostics logs at the WebLogic Server level:

1. Navigate to the WebLogic Server for which you want to configure a logger.
   a. In the navigation pane, expand **WebLogic Domain**.
   b. Expand the domain.
   c. Select the desired server from the list.

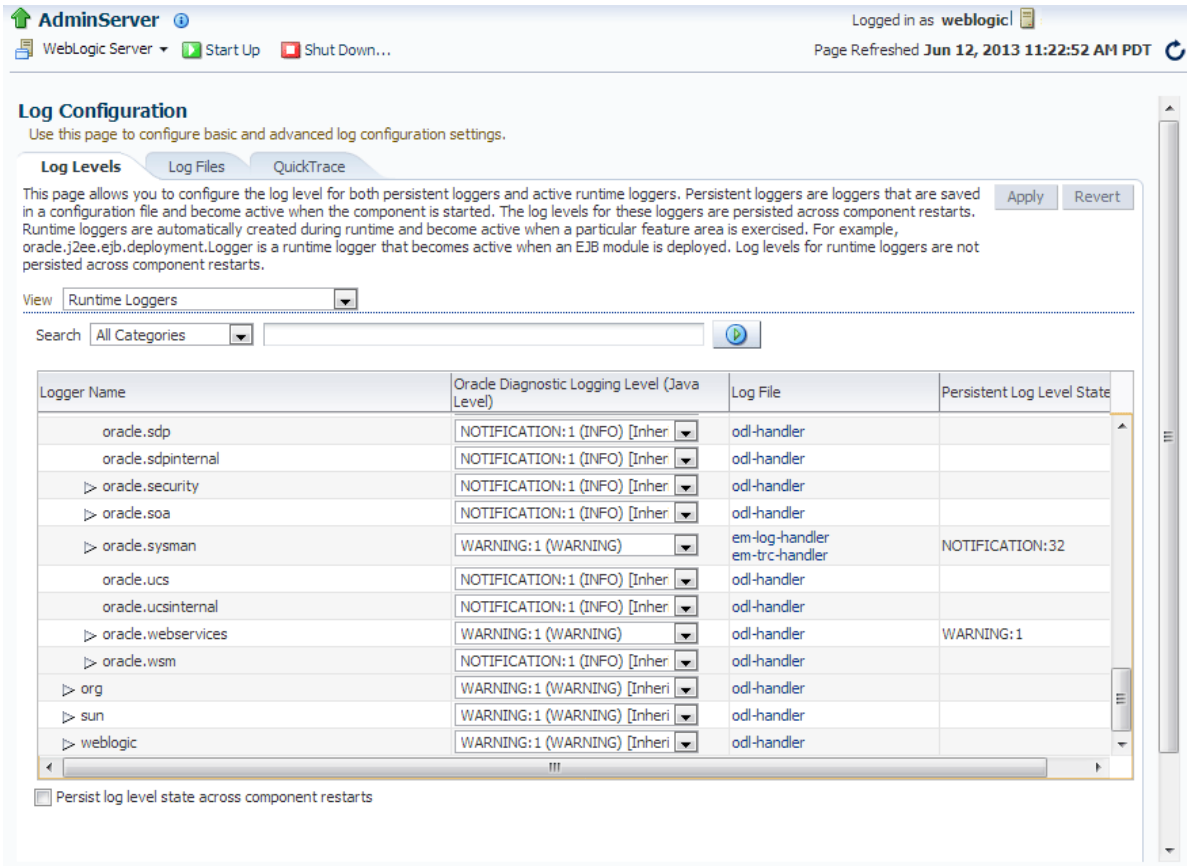
      The WebLogic Server home page is displayed.
2. From the **WebLogic Sever** menu, select **Logs**, then **Log Configuration**.

   The Log Configuration page is displayed.
3. Select the **Log Levels** tab.

   The list of loggers is displayed, as shown in Figure 7-1.

   The Log Levels page shows the name of the logger, the current logging level, which you can edit, and the associated log file (for example, olh-handler). For information about configuring the log files, see "Configuring Log Files for a Web Service".

Figure 7-1 Log Levels Page



4. Expand **Root Logger**.
5. Expand **oracle**.
6. Set the logging level for one or more of the following components:
   - oracle.webservices—web service components.
   - oracle.wsm—OWSM components.

   You can fine tune the logging level by expanding either of the above components and specifying the logging level at the subcomponent level.

   To change the logging level for a logger, navigate to the logger in the Logger Name column and select the desired logging level from the **Oracle Diagnostic Logging Level (Java Level)** drop-down menu.

   For example, select TRACE:32 from the drop-down menu associated with the oracle.wsm logger.

   By default, the logging levels are inherited from the parent and set to NOTIFICATION: 1 (INFO) for the web service and OWSM components and subcomponents.
7. Click **Apply** to store the new logging level.

To set the log level for diagnostic logs at the web service endpoint level:

1. Navigate to the Web Service Endpoint page, as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control".

2. Click the **Configuration** tab.
3. Set the **Logging Level** field to one of the following settings: Severe, Warning, Information, Configuration, Fine, Finer, Finest or NULL.

7.1.1.2 Viewing Diagnostic Logs

You can view the diagnostic log files for an ADF web service endpoint from the Log Messages page.

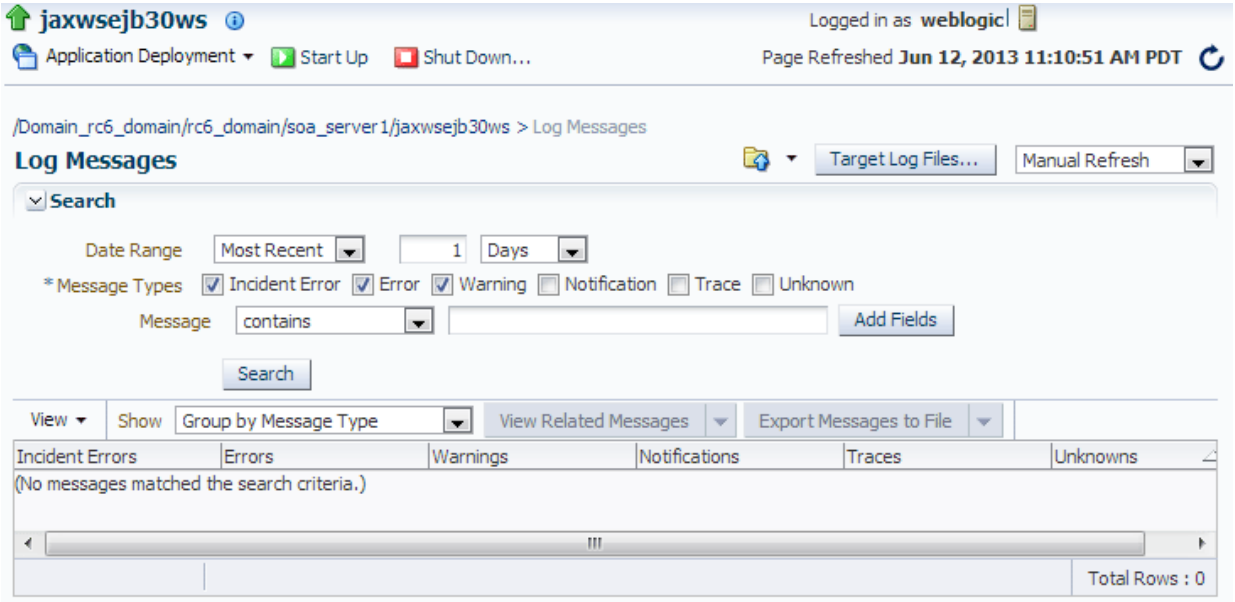
To view diagnostic logs for a web service endpoint:

Navigate to the Web Service Endpoint page, as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control", and in the Quick Links section of the Web Services Endpoint page (top right), click **Diagnostic Log**.

> **Note:**
>
> You can view a summary of all faults incurred by the web services in your application. For more information, see "Overview of Monitoring Web Services".

The Log Messages page is displayed, as shown in the following figure.

Figure 7-2 Log Messages Page



Click on a message in the message area to view more details at the bottom of the page. If desired, you can export a message to a text, XML, or CSV file by selecting the messages on the list and clicking **Export Messages to File**.

You can control the message content displayed using the following controls:

- **Search**—Modify the search criteria. For more information, see "Filtering Diagnostic Logs".
- **View menu**—Select the columns to display in the table. Click on a particular column to sort contents up or down.
- **Show menu**—Group messages by type or ID, or view them in chronological order.
- **View Related Messages**—View messages related to those selected on the list.
-  —Broaden the scope of messages displayed. You can broaden the scope to include all messages for the domain, cluster, application deployment, or WebLogic Server.
- **Refresh menu**—Specify an automatic or manual refresh rate.

To view the contents of a generated log file:

- Click the log file icon associated with a message to view the contents of that log file.
- Click **Target Log Files...** to display the Log Files page and view or download the contents of all generated log files.

For more information, see "Viewing and Searching Log Files" in *Administering Oracle Fusion Middleware*.

7.1.1.3 Filtering Diagnostic Logs

By default, the Log Messages page displays a summary of diagnostic messages logged over the last hour.

To filter diagnostic logs:

1. Filter the messages that are displayed by updating the search criteria using the following fields:
   - **Date Range**—Set the date range to one of the following:
     - Most Recent—Set the amount of time to define the duration.
     - Time Interval—Set the start and end dates to define the interval.
   - **Message Types**—Select the message types that you want to display.
   - **Add Fields**—Add other message fields to your search criteria, such as Message ID, Component, and so on.
2. Click **Search** once you have set the fields, as desired.

   The messages area is updated with the filtered results.

For more information, see "Viewing and Searching Log Files" in *Administering Oracle Fusion Middleware*.

7.1.1.4 Logging OWSM Debug Messages

To debug OWSM, pass one of the following properties when starting WebLogic Server, as required. For more information, see "Starting and Stopping Servers" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

> **Note:**
>
> Enabling one or more of these properties may negatively impact performance for very large messages. When enabled, OWSM creates temporary buffers which will result in additional load on the Java garbage collector.

Table 7-1 Startup Properties for Logging OWSM Debug Messages

Startup Property	Description
`-Dxml.debug.verify=true`	Logs the sequence of bytes produced during a signature verification failure. Verification errors are output to `stderr` and the diagnostic log file when the log level is set to at least ERROR.
`-Dxml.debug.digest=true`	Verifies that the sequence of bytes produced during signature generation canonicalization and signature verification match. Verification errors are output to `stderr` and the diagnostic log file when the log level is set to at least FINE.
`-Dxml.debug.decrypt`	Logs the sequence of bytes produced following a decryption failure before XML parsing. Verification errors are output to `stderr` and the diagnostic log file.

7.1.2 Overview of Message Logs for Web Services

Message logs enable you to access the contents of the SOAP message requests and responses for ADF web services and clients. Messages logs are stored in a log file separate from the diagnostic messages, by default.

The following sections describe how to view and manage message log files:

- Configuring Message Logs
- Viewing Message Logs
- Filtering Message Logs

7.1.2.1 Configuring Message Logs

You configure message logs for a web service or client in one of the following ways:

- Attach a policy that contains a logging assertion to the web service or client.

  There is one predefined logging assertion template: `oracle/security_log_template`, described in "oracle/security_log_template" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*. This template is configured to log the entire SOAP message for the web service request and response. By default, all predefined web service security policies use this logging assertion to capture the entire SOAP message before and after the primary security assertion is executed. By default, the log assertion is not enforced. You must enable it in order for the SOAP message to be logged in message logs, as described in "Enabling or Disabling Assertions Within a Policy" in *Securing Web Services and Managing Policies with Oracle Web Services*

Manager. It is recommended that the logging assertion be enabled for debugging and auditing purposes only.

- Attach the `oracle/log_policy` policy to the web service or client. For more information, see "oracle/log_policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
- Create your own logging policy or assertion template to further refine the elements of the SOAP message that are logged for the web service request and response.

  For example, you may wish to view only the SOAP body of the request message. To create a new policy, following the procedure described in "Creating a New Web Service Policy" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*. You may wish to create a copy of the `oracle/security_log_template` assertion template and configure it for use in the new policy. For more information about creating a new assertion template, see "Cloning an Assertion Template" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

7.1.2.2 Viewing Message Logs

You can view the message log files for an ADF web service endpoint from the Log Messages page.

To view message logs for a web service endpoint:

Navigate to the Web Service Endpoint page, as described in "Introduction to Viewing Details for a Web Service Endpoint Using Fusion Middleware Control", and in the Quick Links section of the Web Services Endpoint page (top right), click **Message Logs.**

The Log Messages page is displayed, similar to Figure 7-2. For more details about the contents of the Log Messages page, see "Viewing Diagnostic Logs".

7.1.2.3 Filtering Message Logs

By default, the Log Messages page displays a summary of SOAP messages logged over the last hour. You can filter the messages that are displayed by updating the search criteria. The process is the same as for diagnostic logs; for more information, see "Filtering Diagnostic Logs".

By default, the Component and Module message fields are included as part of the Search criteria for message logs. The Component field is set to the WebLogic Server name; the Module field is set to oracle.wsm.msg.logging, which is the name of the message logging component.

7.1.3 Sample Logs

The following sections provide excerpts from sample logs, demonstrating how to diagnose specific problems using the log entries.

- Sample Log: OWSM Policy Manager Not Available
- Sample Log: Security Keystore Not Configured
- Sample Log: Certificate Not Available

7.1.3.1 Sample Log: OWSM Policy Manager Not Available

The following sample log excerpt indicates that the OWSM Policy Manager is down. To resolve this issue, restart the wsm-pm application, as described in "Starting and Stopping Applications" in *Administering Oracle Fusion Middleware*.

```
2009-02-16 16:21:28,029 [[ACTIVE] ExecuteThread: '4' for queue:
 'weblogic.kernel.Default (self-tuning)']
 ERROR policymgr.PolicyManagerModelBean logp.251 -
 Service lookup failed with URL:t3://host.example.com:7001/wsm-pm
 oracle.wsm.policymanager.PolicyManagerException: WSM-02118 :
 The query service cannot be created.
...
```

7.1.3.2 Sample Log: Security Keystore Not Configured

The following sample log excerpt indicates that an OWSM security policy with message protection was applied, but the keystore was not configured. To resolve this security fault, configure the keystore, as described in "Configuring Keystores for Message Protection" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

```
Feb 16, 2009 5:29:56 PM oracle.wsm.common.logging.WsmMessageLogger logSevere
 SEVERE: The specified Keystore file /scratch/sbollapa/stage131/user_projects/
domains/sai131_domain/config/fmwconfig/default-keystore.jks
 cannot be found; it either does not exist or its path is not included in the
 application classpath.
 Feb 16, 2009 5:29:56 PM oracle.wsm.common.logging.WsmMessageLogger logSevere
 SEVERE: Keystore is not properly configured in jps config.
 Feb 16, 2009 5:29:56 PM oracle.wsm.common.logging.WsmLogUtil log
 SEVERE: failure in OWSM Agent processRequest, category=security,
 function=agent.function.client, application=default, composite=pe3test3,
 modelObj=Service1, + policy=null, policyVersion=null, assertionName=null
 oracle.wsm.common.sdk.WSMException: WSM-00101 : The specified Keystore file
 /scratch/sbollapa/stage131/user_projects/domains/sai131_domain/config/fmwconfig/
default-keystore.jks cannot be found;
 it either does not exist or its path is not included in the application classpath.
...
```

7.1.3.3 Sample Log: Certificate Not Available

The following sample log excerpt indicates that an OWSM security policy with message protection was applied that required a security certificate that was not available in the keystore. To resolve this security fault, configure the keystore with a certificate, as described in "Obtaining a Trusted Certificate and Importing it into the Keystore" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

```
[2009-04-15T04:07:02.821-07:00] [jrfServer] [ERROR] [WSM-000062]
 [oracle.wsm.resources.security] [tid: [ACTIVE].ExecuteThread: '0' for
queue: 'weblogic.kernel.Default (self-tuning)'] [userId: <anonymous>]
 [ecid: 0000I2dTFG7DScT6uBe9UH19tRyv000000,0:1] [WEBSERVICE_PORT.name:
 NonCAAsCAMessageProtectionPolicyPort] [APP: jaxwsservices]
 [J2EE_MODULE.name: NonCAAsCAMessageProtectionPolicy] [WEBSERVICE.name:
 NonCAAsCAMessageProtectionPolicyService] [J2EE_APP.name: jaxwsservices]
 [arg: oracle.wsm.security.SecurityException: WSM-00062 :
 The path to the certificate used for the signature is invalid.]
```

```
[2009-04-15T04:07:02.810-07:00] [jrfServer] [NOTIFICATION] []
 [oracle.wsm.security.policy.scenario.processor.Wss11X509TokenProcessor]
 [tid: [ACTIVE].ExecuteThread: '0' for queue: 'weblogic.kernel.Default
 (self-tuning)'] [userId: <anonymous>]
[ecid: 0000I2dTFG7DScT6uBe9UH19tRyv000000,0:1]
[WEBSERVICE_PORT.name: NonCAAsCAMessageProtectionPolicyPort]
[APP: jaxwsservices] [J2EE_MODULE.name: NonCAAsCAMessageProtectionPolicy]
 [WEBSERVICE.name: NonCAAsCAMessageProtectionPolicyService] [J2EE_APP.name:
 jaxwsservices] Certificate path validation failed for signing certificate

[2009-04-15T04:07:02.821-07:00] [jrfServer] [ERROR] [WSM-00006]
 [oracle.wsm.resources.security] [tid: [ACTIVE].ExecuteThread: '0' for queue:
 'weblogic.kernel.Default (self-tuning)'] [userId: <anonymous>]
[ecid: 0000I2dTFG7DScT6uBe9UH19tRyv000000,0:1] [WEBSERVICE_PORT.name:
 NonCAAsCAMessageProtectionPolicyPort] [APP: jaxwsservices]
[J2EE_MODULE.name: NonCAAsCAMessageProtectionPolicy] [WEBSERVICE.name:
 NonCAAsCAMessageProtectionPolicyService] [J2EE_APP.name: jaxwsservices]
[arg: oracle.wsm.security.SecurityException: WSM-00062 : The path to the
 certificate used for the signature is invalid.] Error in receiving the request:
 oracle.wsm.security.SecurityException: WSM-00062 : The path to the certificate
 used for the signature is invalid.
```

7.2 Configuring Log Files for a Web Service

To further organize your logging data, you can configure the log files for a web service.

You can configure log files for SOA and ADF services.

The following table defines the default log files that are relevant to OWSM.

Table 7-2 Default Log Files for OWSM

Default Log File	Description
odl-handler	Logs general diagnostic data for the Java EE components in the server.
owsm-message-handler	Logs SOAP messages as per OWSM logging policies.

The following procedure describes how to set the log level for diagnostic logs at the WebLogic Server and web service endpoint levels.

For more information about using Fusion Middleware Control or WLST to set the log levels, see "Setting the Level of Information Written to Log Files" in *Administering Oracle Fusion Middleware*.

To configure the log files for a web service:

1. Navigate to the WebLogic Server for which you want to configure a logger.
   a. In the navigation pane, expand **WebLogic Domain** to view the domain name.
   b. Expand the domain to see the list of servers.
   c. Select the desired server from the list.

      The WebLogic Server home page is displayed.
2. From the **WebLogic Server** menu, select **Logs > Log Configuration**.

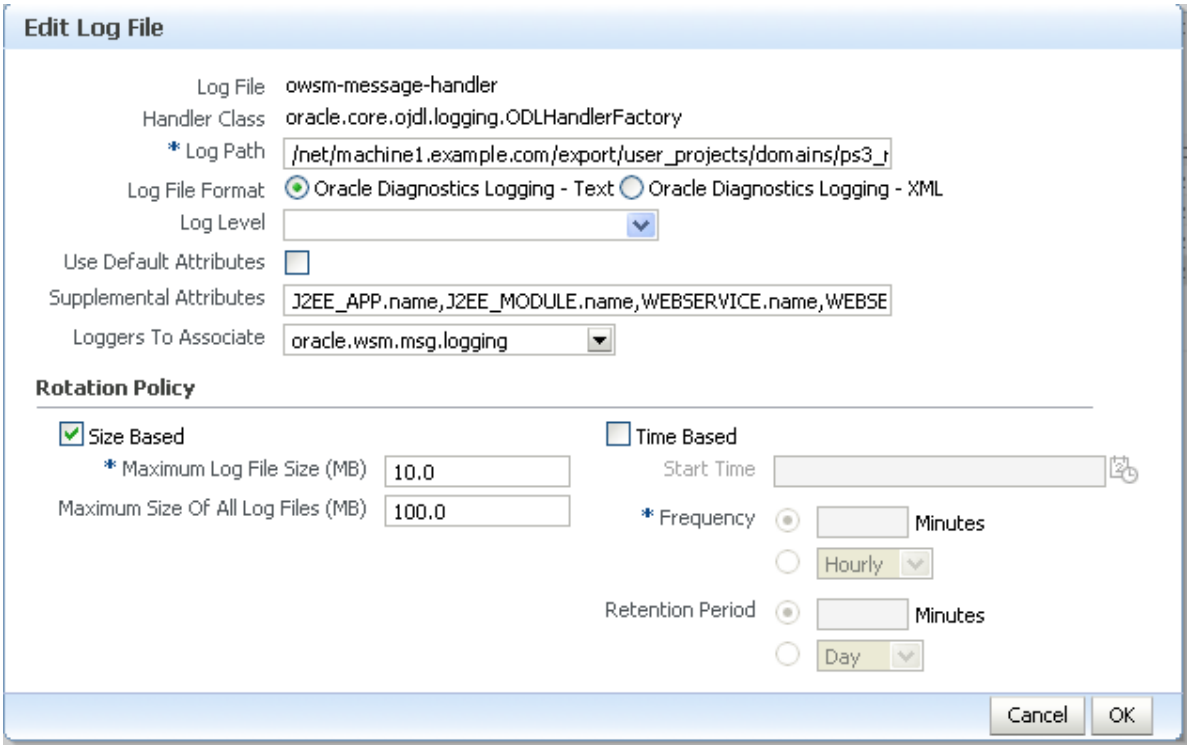
   The Log Configuration page is displayed.
3. Select the **Log Files** tab.

The current list of log files is displayed. The Log Configuration page shows the currently configured log path, file format, and rotation policy.

4. If you wish to edit the log policy configuration, select the log file in the list and click **Edit Configuration . . .**.

   The Edit Log File page is displayed.

Figure 7-3 Edit Log File Page



5. Edit the log file information, as required.

Table 7-3 Fields in Edit Log File Page

Field	Description
Log Path	Path to the log file. This field is required.
Log File Format	Format of the log file. Valid values are text or XML.
Log Level	Default log level for the logger. Select a log level from the list. Valid values include: • INCIDENT_ERROR:1 (SEVERE+100) • ERROR:1 (SEVERE) • WARNING:1 (WARNING) • NOTIFICATION:1 (INFO) • NOTIFICATION:16 (CONFIG) • TRACE:1 (FINE) • TRACE:16 (FINER) • TRACE:32 (FINEST)
Use Default Attributes	Flag that specifies whether to use default attributes for the logger.

Table 7-3 (Cont.) Fields in Edit Log File Page

Field	Description
Supplemental Attributes	Supplemental attributes required.
Loggers to Associate	Components to associate with the logger.
Rotation Policy	Specify whether you wish to rotate log files based on file size of length of time. For more information, see "Configuring Log File Rotation" in *Administering Oracle Fusion Middleware*. If Size Based is selected as the rotational policy, Maximum Log Files Size is a required field. If Time Based is selected as the rotational policy, Frequency is a required field.

6. Click **OK** to edit the log file configuration.

8

Managing Application Migration Between Environments

To migrate web service applications independently between environments, such as from test to production, or in a scaled clustered environment, you must export the policies and the deployment configuration information to the new environment so that you can deploy the application. Depending on your configuration, you may also need to migrate policy configuration artifacts and policy assertion templates.
This chapter includes the following sections:

For information about moving from a test environment to a production environment, see "Moving from a Test to a Production Environment" in *Administering Oracle Fusion Middleware*.

8.1 Introduction to Web Service Application Migration

A deployment descriptor is an XML file that contains the basic deployment configuration for an application.

For WebLogic Server and Java EE web service applications, you create a deployment plan that contains the necessary deployment descriptors for deploying the application in a new environment.

For ADF Business Components and WebCenter services, however, run-time policy changes are persisted in proprietary deployment descriptor (PDD) files: `oracle-webservices.xml` and `oracle-webservices-client.xml`. Because these files are not included in the WebLogic deployment plan or exported with any other deployment descriptors, you must export and import these PDD files separately. You must also export and import these PDD files separately if you are scaling your application in a clustered environment.

Note that the following Oracle Infrastructure web services components provide different configuration management mechanisms.

- For a SOA composite, web services and OWSM configurations are persisted in a composite.xml file which is included in a configuration plan used for deployment configuration. The SOA framework provides its own mechanism for composite services and configuration lifecycle and synchronizations.

- ADF Web Service data control configuration stores connection details for WebCenter services in a connections.xml file and all post-deployment changes as customizations in the Metadata Services (MDS) repository.

8.2 Migrating a Web Service Application from a Development or Test Environment to a Production Environment

The general steps for migrating a web service application from a development or test environment to a production environment are as follows:

1. Install and configure the production environment with the components that you need.
2. Migrate security information, such as users and groups, the identity and policy stores, and credentials. For more information, see "Overview of Migrating Policy Configuration".
3. Migrate policies and deployment configuration data as required. For more information, see "Migrating Policies". Modify any information that is specific to the new environment such as host name or ports.
4. Deploy the applications in the new environment.

For information about migrating Fusion Middleware applications between environments, see "Advanced Administration: Expanding Your Environment" in *Administering Oracle Fusion Middleware*.

8.3 Creating and Migrating a Policy Horizontally Through the Different Stages

The following steps describe a typical scenario for how to create a policy and migrate the policy horizontally through the different stages of the application development and deployment cycles.

1. Use Oracle Enterprise Manager Fusion Middleware Control to create a policy.

   For more information, see "Creating and Editing Web Service Policies" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.
2. Export the policy to a zip archive.

   For more information, see "Migrating Policies".
3. Extract the contents of the `META-INF` directory in the zip archive to the policy store location in the Oracle JDeveloper environment. For more information, see "Exporting Policies from the OWSM Repository for Use in JDeveloper" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*
4. Create a web service in Oracle JDeveloper and attach the policy to the web service.

   For more information, see "Developing and Securing Web Services" in *Developing Applications with Oracle JDeveloper*.
5. Deploy the web service to the staging server, and test the web service.

For more information, see "Developing and Securing Web Services" in *Developing Applications with Oracle JDeveloper*.

6. Import the policy to the production server environment.

   For more information, see "Migrating Policies".

7. Migrate the following information, as required:
   - Policy configuration. See "Overview of Migrating Policy Configuration".
   - Assertion templates. See "Migrating Assertion Templates".

8. Deploy the application into the production environment, and test the web service.

   See Deploying Web Service Applications and Testing Web Services.

8.4 Migrating Policies

You can export one or more user-created policies to an archive file using Fusion Middleware Control. You can then import the archive to move it to another repository.

> **Note:**
>
> Read-only documents, such as predefined policies and assertion templates, will not be imported or exported using either Fusion Middleware Control or WLST because they will already be present in the target environment.

For details about exporting and importing user-created policies using Fusion Middleware Control, refer to the following topics in *Securing Web Services and Managing Policies with Oracle Web Services Manager*:

- Exporting Web Service Policies
- Importing Web Service Policies

Alternatively, you can use the `exportWSMRepository` and `importWSMArchive` WLST commands to export and import the policies.

To migrate policies using WLST commands:

1. Export the OWSM policies to a supported archive file, such as a zip file, using the `exportWSMRepository` command.

   For example, to export all user-created OWSM policies named in the test/ directory to an archive named `policies.zip`, enter the following:

```
wls:/jrfServer_domain/serverConfig> exportWSMRepository('/tmp/policies.zip',
['policies:test/%'])

Exporting "/policies/test/
wss11_x509_token_with_message_protection_service_policy_test"
Exporting "/policies/test/wss_username_token_over_ssl_service_policy_Test"
Successfully exported "2" documents.
```

2. Optionally, you can edit the archive after it has been created. If, for example, you do not want to migrate all the policies to the new environment, you can manually remove them from the archive.

3. Move the archive to the new machine. Ensure that the OWSM Policy Manager is deployed on the new machine.
4. Import the OWSM policies using the `importWSMArchive` command. For example, to import the policies exported in the previous step:

```
wls:/jrfServer_domain/serverConfig> importWSMArchive('/tmp/policies.zip')

Importing "META-INF/policies/test/
wss11_x509_token_with_message_protection_service_policy_test"
Importing "META-INF/policies/test/
wss_username_token_over_ssl_service_policy_Test"
Successfully imported "2" documents
```

For more information about these WLST commands, see "Web Services Custom WLST Commands" in *WLST Command Reference for Infrastructure Components*.

8.5 Overview of Migrating Policy Configuration

Migration of an application between environment requires migration of configuration artifacts for OWSM policies, such as keystores, users and groups.

The following sections describe how to migrate the configuration artifacts for OWSM policies. This section includes the following topics:

- Migrating Keystores
- Migrating Users and Groups
- Migrating Credentials
- Migrating Oracle Platform Security Services Application and System Policies
- Migrating Oracle Platform Security Services Configuration
- Migrating SSL
- Migrating Kerberos Configuration

8.5.1 Migrating Keystores

If you are using message protection policies, you need to migrate your keystores.

To migrate keystores:

1. Manually copy your keystores to the new environment.

   For Java SE applications, copy the keystore to a user-defined location. For Java EE applications, copy the keystore to the same directory as the `jps-config.xml` file, namely `DOMAIN_HOME/config/fmwconfig`.
2. By default, the keystore is named default-keystore.jks. If you have renamed the keystore, you must configure the keystore name in the Oracle Platform Security Services keystore service instance.

For information about configuring the keystore, see "Configuring Keystores for Message Protection" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

To migrate keystores with Keystore Service:

1. Export the keystore to a file with the `exportKeyStore` command.

2. Import the file to the new keystore with the `importKeyStore` command.

For information about using the keystore migration commands in KSS, see "Managing Keys and Certificates" in *Securing Applications with Oracle Platform Security Services*.

8.5.2 Migrating Users and Groups

Users and groups are maintained as part of the WebLogic Server security realm. To migrate users and groups in embedded LDAP, you can migrate the data using either the Oracle WebLogic Administration Console or WLST.

For a complete description of the steps required, see "Migrating Security Data" in *Administering Security for Oracle WebLogic Server*.

To migrate users and groups in an LDAP store, there is no migration path. You need to recreate the users and groups and specify the assignments in the LDAP store in the new environment. See "Configuring Authentication Providers" in *Administering Security for Oracle WebLogic Server*.

8.5.3 Migrating Credentials

There are two types of credentials maintained in the credential store that you may need to migrate:

- Username and password
- Keystore and encryption key passwords

The migration steps are described in the sections below.

8.5.3.1 Migrating Username and Password

If users are stored in an embedded LDAP and migrated, as described in "Migrating Users and Groups", then you simply migrate the existing credentials to the new credential store. For a complete description of the steps required, see "Migrating Security Data" in *Administering Security for Oracle WebLogic Server*.

If users are stored in an LDAP store, there is no automated migration path. You need to recreate the credentials in the credential store. For more information about configuring credentials, see "Configuring the Credential Store" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

8.5.3.2 Migrating Keystores and Encryption Key Passwords

You can migrate keystores and encryption key passwords manually using the procedure described in "Migrating Credentials Manually" in "Deploying Secure Applications" in *Securing Applications with Oracle Platform Security Services*.

8.5.4 Migrating Oracle Platform Security Services Application and System Policies

If your web service uses authorization policies, you must migrate the Oracle Platform Security Services application and system policies that grant permissions.

For more information, see "Migrating with the Script migrateSecurityStore" in "Configuring the OPSS Security Store" in *Securing Applications with Oracle Platform Security Services*.

8.5.5 Migrating Oracle Platform Security Services Configuration

There is no automated migration path for Oracle Platform Security Services configuration. You must recreate the configuration in the new environment.

There are three types of configurations in the Oracle Platform Security Services that you may need to recreate:

- SAML trusted assertion issuer names (applicable for all SAML policies).

  If you use the default configuration for SAML trusted issuer configuration, then no migration is required. For information about configuring SAML in the new environment, see "Configuring the SAML and SAML2 Login Modules Using Fusion Middleware Control" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Keystore locations and CSF key configuration for keystore and keystore password (applicable for message protection policies only).

  If you use the default configuration for keystores, then no migration is required. For information about configuring keystores in the new environment, see "Configuring Keystores for Message Protection" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

- Keytab location and service principal name (applicable to Kerberos policy).

  For information about configuring the keytab location and service principal name in the new environment, see the following topics in *Securing Web Services and Managing Policies with Oracle Web Services Manager*:

  – Configuring the SAML and SAML2 Login Modules Using Fusion Middleware Control

  – Configuring the Kerberos Login Module Using Fusion Middleware Control

8.5.6 Migrating SSL

There is no automated migration path for SSL configuration. You must configure SSL keystores and settings in the new environment.

For more information about configuring SSL keystores and settings in the new environment, see "Configuring Keystores for SSL" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

8.5.7 Migrating Kerberos Configuration

To migrate the Kerberos configuration:

1. Copy the Kerberos configuration file to the new environment, matching the directory structure. The Kerberos configuration file is located in the following locations, based on your operating system:
   - **UNIX**: `/etc/krb5.conf`
   - **Windows**: `C:\windows\krb5.ini`
2. Initialize the ticket cache with the correct credentials.

   For more information, see "Configuring Kerberos Tokens" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

8.6 Migrating Assertion Templates

You can export individual assertion templates from Oracle Enterprise Manager Fusion Middleware Control. You can then copy the policy to a directory or import the policy to move it to another repository.

For details about exporting and importing assertion templates, see "Managing Policy Assertion Templates" in *Securing Web Services and Managing Policies with Oracle Web Services Manager*.

9

Viewing, Registering, and Publishing Web Services

This chapter describes how to register web services and sources in Enterprise Manager from a WSIL or UDDI registry. It also describes how to publish registered resources to the UDDI registry.
This chapter includes the following sections:

- Introduction to Registering Web Services and Sources
- Introduction to Publishing Web Services to UDDI

9.1 Introduction to Registering Web Services and Sources

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.

Oracle Enterprise Manager Fusion Middleware Control can register web services that are published in WSIL documents and UDDI v3 registries. Any service that is available in a WSIL document or a UDDI v3 registry can be registered within Enterprise Manager.

You can also register meta information, or a profile, for sources of services to help you manage your registered services within Enterprise Manager. Once you register a source and assign it a logical name, you do not need to specify connectivity information, such as a URL for a WSDL, in the future. A domain can have multiple registered sources, and each registered source can have multiple registered services. Once you register a source, you can easily look up services that you can register to the source.

Service names and corresponding WSDLs must be unique within a registered single source. Once you have registered a service, an attempt to register another service with the same name, or a different name but the same WSDL URL as another service, is not valid.

Once you register a web service, you can later, more conveniently, reference the service from a selection list within Enterprise Manager. For example, when testing a web service as described in Testing Web Services, instead of specifying a WSDL, you can click the **Search** icon and then select the WSDL from the list of registered services.

This section includes the following topics:

- Understanding UDDI Basics
- Understanding WSIL Basics
- Viewing Registered Sources and Web Services

- Registering a Source
- Registering Web Services from a UDDI Source
- Registering Web Services from a WSIL Source
- Deleting a Web Service or Web Service Source

9.1.1 Understanding UDDI Basics

Universal Description Discovery & Integration (UDDI) is an industry initiative that aims to enable businesses to quickly, easily, and dynamically find and carry out transactions with one another. A populated UDDI registry contains cataloged information about businesses; the services that they offer; and communication standards and interfaces they use to conduct transactions.

The owners of web services publish them to the UDDI registry. Once published, the UDDI registry maintains pointers to the web service description and to the service. The UDDI allows clients to search this registry, find the intended service, and retrieve its details. These details include the service invocation point as well as other information to help identify the service and its functionality.

9.1.2 Understanding WSIL Basics

WSIL defines an Extensible Markup Language (XML) format for referencing web service descriptions. 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.

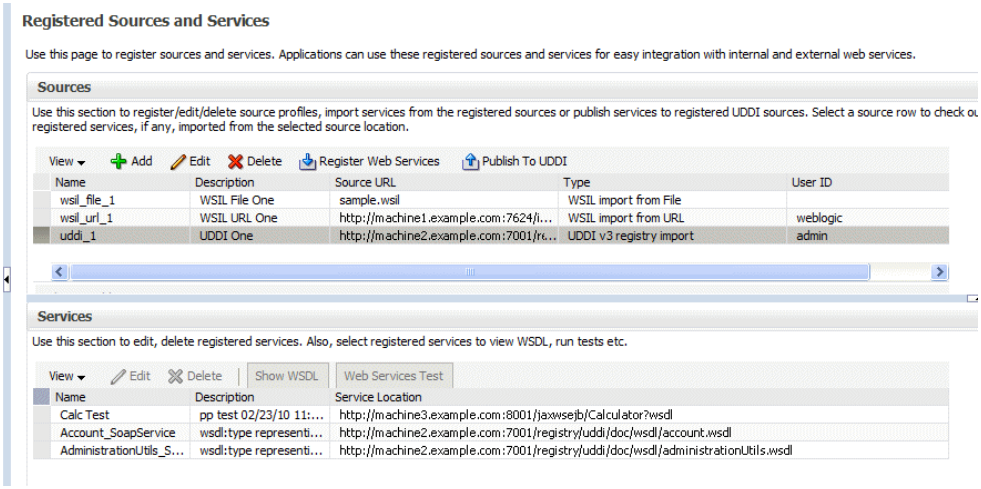
The WSIL model decentralizes web service discovery. In contrast to UDDI registries, which centralize information on multiple business entities and services, WSIL makes it possible to provide web service description information from any location. Unlike UDDI, WSIL is not concerned about business entity 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.

9.1.3 Viewing Registered Sources and Web Services

Follow the steps in this section to view and edit a registered source and web service.

1. In the navigation pane, expand **WebLogic Domain** to show the domain in which you want to view the registered sources and web services.
2. Select the domain.
3. From the **WebLogic Domain** menu, select **web Services** then **Registered Services**. The Registered Sources and Services page appears, as shown in Figure 9-1.

Figure 9-1 Viewing Registered Sources and Services



In the Sources table, you can view the following information about each registered source:

- Name—Logical name for the source
- Description—Description of the source
- Source URL—Location of the source in URL format
- Type—Source type: UDDI v3 registry import, WSIL import from file, WSIL import from URL
- User ID—User ID for the external source

You can customize the information that is displayed using the **View** menu. From this section of the page, you can also add new sources, edit or delete sources, register web services for a source, and publish a web service from a source to a predefined UDDI registry.

4. Select a source in the Sources table.

   Each registered source can have multiple registered services. In the Services table, you can view the following information about the registered services imported from the selected source location:

   - Name—Name of the registered service
   - Description—Description of the service
   - Service location—Location of the service in URL format

   You can customize the information that is displayed using the **View** menu. You can also display the WSDL for the selected service and test the selected web service.

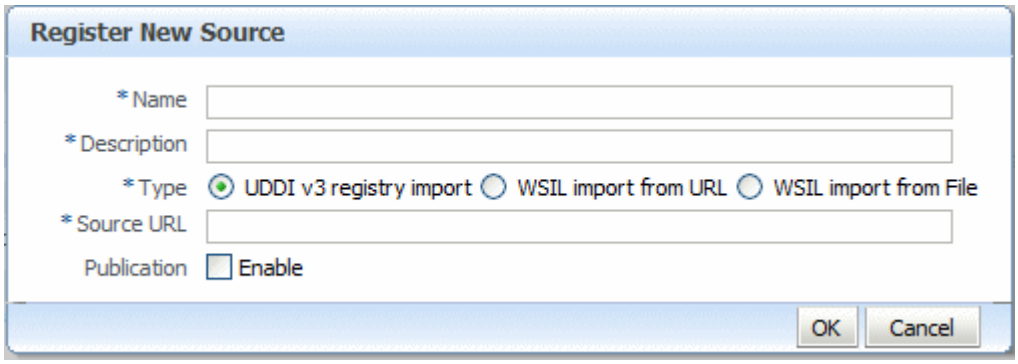
9.1.4 Registering a Source

You can register web service sources of the following types: UDDI v3 registry import, WSIL import from URL, or WSIL import from file.

To register a source:

1. In the navigation pane, expand **WebLogic Domain** to show the domain in which you want to register a web service source.
2. Select the domain.
3. From the **WebLogic Domain** menu, select **Web Services** then **Registered Services**. The Registered Sources and Services page appears, as shown in Figure 9-1.
4. Click **Add** to register a new source. The Register New Source page appears, as shown in Figure 9-2.

Figure 9-2 Register New Source Page



5. Enter the following information for the new source.
   - **Name**—A logical name for the source.
   - **Description**—A description of the source.
   - **Type**—choose from one of three options: **UDDI v3 registry import**, **WSIL import from URL**, or **WSIL import from File**

     Additional information that you need to enter differs based on the option you select.
6. If you selected **UDDI v3 registry import**, enter the following information:
   - In the **Source URL** field, enter the UDDI inquiry URL, for example, `http://somehost/uddi/inquiry`.
   - To allow the services to be published to a UDDI source (which is an external UDDI registry), select the **Enable** box and complete the fields as follows:
     - In the **Publication URL** field, enter URL location of the registry to which you want to publish the service.
     - In the **Security URL** field, enter the URL location of the security port required to access the registry.
     - In the **User ID** and **Password** fields, enter the security credentials required to access the registry.
7. If you selected **WSIL import from URL**, enter the following information:
   - In the **Source URL** field, enter the location of the WSIL in URL form.
   - If a username and password are required to access the WSIL, select the **Enable** box in the **Basic Authorization** field. In the **User ID** and **Password** fields, enter the username and password.

8. If you selected **WSIL import from File**, click **Browse** (next to the **WSIL File** field) to select the WSIL file to be imported.
9. Click **OK** to register the source.

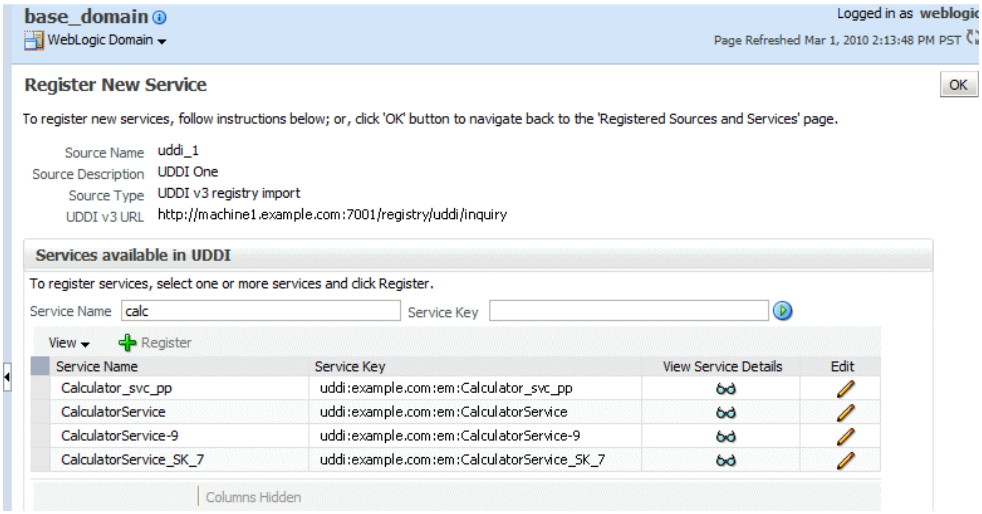
9.1.5 Registering Web Services from a UDDI Source

Follow the steps in this section to register web services from a registered UDDI source.

1. In the navigation pane, expand **WebLogic Domain** to show the domain in which you want to register a web service.
2. Select the domain.
3. From the **WebLogic Domain** menu, select **Web Services** then **Registered Services**. The Registered Sources and Services page displays, as shown in Figure 9-1.
4. Select the UDDI source from which you want to register services. Note that the Type for a UDDI source is specified as UDDI v3 registry import.
5. Select **Register Web Services**.

   The Register New Service page displays, as shown in Figure 9-3.

Figure 9-3 Register New Service from UDDI Source



The Register New Service page displays the source information, in read-only format, and a list of the services that are available in UDDI that you can register.

You can filter the list of available services that are displayed using the **Service Name** and **Service Key** fields. For example, to find calculator services, enter `calc` in the **Service Name** field. Only services that contain the `calc` string, such as calculator services are displayed. The search is not case-sensitive.

In the Services available in UDDI section of the page, you can view service details from UDDI for each service in the table by clicking the **View Service Details** icon. The Service Details from UDDI window displays information about the service

such as the Service Name, Service Description, Service WSDL, Service Key, Business Key, and Service Location, among others.

You can edit the details of a service by clicking the **Edit** icon, which allows you to change the name and description of the selected service.

6. In the Services available in UDDI section of the page, select the service or services that you want to register from the source and click **Register**.

   A confirmation message displays indicating that the service was registered successfully.

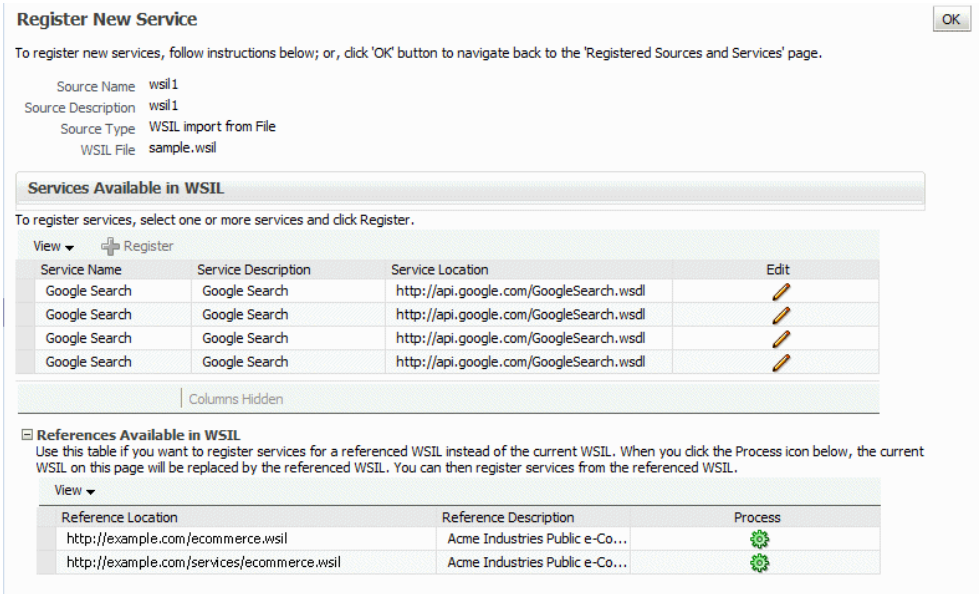
9.1.6 Registering Web Services from a WSIL Source

Follow the steps in this section to register web services from a registered WSIL source.

1. In the navigation pane, expand **WebLogic Domain** to show the domain in which you want to register a web service.
2. Select the domain.
3. From the **WebLogic Domain** menu, select **Web Services** then **Registered Services**. The Registered Sources and Services page displays, as shown in Figure 9-1.
4. Select the WSIL source from which you want to register services. Note that the Type for a WSIL source is specified as either WSIL import from File or WSIL import from URL.
5. Select **Register Web Services**.

   The Register New Service page displays, as shown in Figure 9-4.

Figure 9-4 Register New Service from WSIL Source



The Register New Service page displays the Source information, in read-only format, and a list of available services, if any, in the WSIL that you can register,

shown in the Services Available in WSIL table. If there are any WSIL references in the WSIL, they are listed in the References Available in WSIL table.

In the Services Available in WSIL table, you can edit the details of a service by clicking the **Edit** icon, which allows you to change the name and description of the selected service.

6. To register a service available from the current WSIL, select the service in the Services Available in WSIL table and click **Register**.

   A confirmation message displays indicating that the service was registered successfully.

7. If the current WSIL also references other WSIL URLs or references, expand **References Available in WSIL** to display them. You can register the referenced web services as well.

   To register a service from a referenced WSIL instead of the current WSIL, click the **Process** icon for the reference in the References Available in WSIL table.

   If the WSIL parses successfully, a new source is registered and the current WSIL source is replaced by the referenced WSIL. The services available in the referenced WSIL source are listed in the Services Available in WSIL table. You can then register services from the referenced WSIL.

   > **Note:**
   >
   > For each new source created, _n is appended to the parent source name. For example, if the parent source name is `wsil_file_1`, then referenced new sources are named `wsil_file_1_1`, `wsil_file_1_2`) with source type WSIL URL. The new sources are listed in the Sources table in the Registered Sources and Services page.

   If the WSIL does not parse successfully, an error message displays. Usually, in such cases, the system successfully registers the new source for the selected WSIL reference. However, because the system could not parse the WSIL document, the error message displays. Close the error dialog and click **OK** to return to the Registered Sources and Services page.

   WSIL parsing can fail if the reference is bad or it needs authorization credentials. You can enable authorization for the WSIL source as described in "Registering a Source".

> **Note:**
>
> When the system fails to retrieve web services from a registered source, because of connection or other failures, the Register New Service page is displayed with read only information for the source, but does not show any web services. In such cases, click **OK** in the error dialog, if an error dialog is displayed, then click **OK** in the Register New Service page to return to the Registered Sources and Services page. To troubleshoot, you can then view the registered sources through other means. For example, if the source is a:
>
> - WSIL URL source, copy the URL to a browser address bar to view its contents.
> - WSIL file source, examine the WSIL file using an XML editor.
> - UDDI source, try to access the UDDI registry directly to investigate.
>
> You can also review any related Enterprise Manager error logs.

9.1.7 Deleting a Web Service or Web Service Source

Follow the steps in this section to delete a web service or a web service source.

1. In the navigation pane, expand **WebLogic Domain** to show the domain in which you want to delete a web service.
2. Select the domain.
3. From the **WebLogic Domain** menu, select **Web Services** then **Registered Services**. The Registered Sources and Services page displays, as shown in Figure 9-1.
4. Do one of the following:
   - To delete a source, select the source from the Sources table and click **Delete**.

     A confirmation message displays. Click **OK** to delete the source.
   - To delete a service from a source, select the source in the Sources table.

     The registered web services are displayed in the Services table.

     Select the service to be deleted from the Services table and click **Delete**.

     A confirmation message displays. Click **OK** to delete the service.

9.2 Introduction to Publishing Web Services to UDDI

You can publish web services to UDDI from a registered UDDI source and from the web services summary page for ADF and Java EE applications. Registered UDDI sources are listed in the Registered Sources and Services page, which includes all sources and services registered in a domain. The Web Services summary page lists the web services in an application.

> **Note:**
>
> You need to use a proxy to publish a service to UDDI, since this requires access to URLs outside of your firewall. For more information about the required proxy settings, see "Configuring the Proxy Server for UDDI".
>
> If your services are already in Oracle Enterprise Repository (OER) then you should use the OER Exchange Utility to publish those services to Oracle Service Registry.

The following procedures describe how to publish web services to UDDI.

- "Publishing a Web Service to UDDI from a Registered Source"
- "Publishing a Web Service to UDDI from an Application"
- "Configuring the Proxy Server for UDDI"

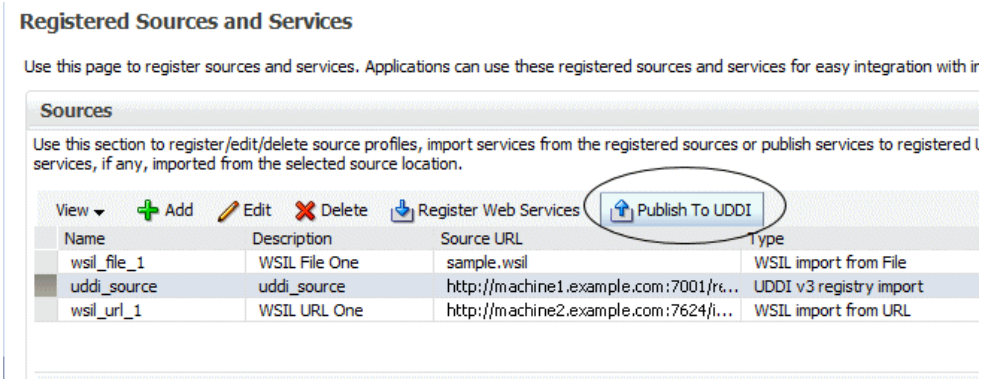
9.2.1 Publishing a Web Service to UDDI from a Registered Source

Registered UDDI sources are listed in the Registered Sources and Services page, which includes all sources and services registered in a domain.

To publish a web service to UDDI from a registered source:

1. Navigate to the Registered Sources and Services page as described in "Viewing Registered Sources and Web Services".
2. Select the source row in the Sources table and then **Publish to UDDI.**

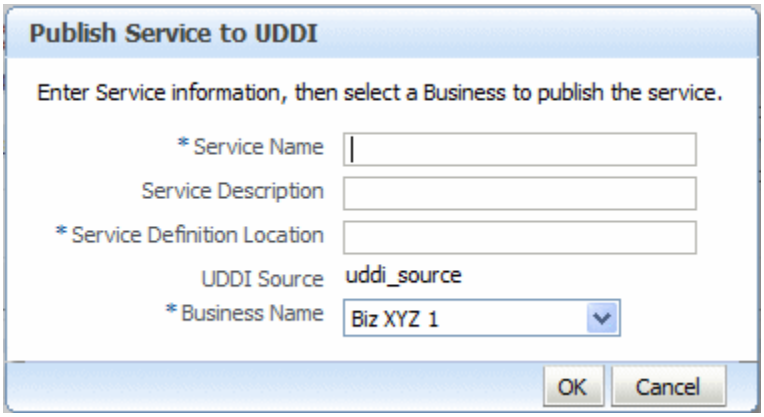
Figure 9-5 Registered Sources and Services Page with Publish to UDDI Selected



3. In the Publish Service to UDDI window, enter the information about the service to be published:
   - **Service Name** is the name of the web service to be published to the UDDI registry. This field is required.
   - **Service Description** is a description of the selected web service.
   - **Service Definition Location** is the URL location of the service definition. This field is required.

- **UDDI Source** is the name of the UDDI source from which the service is to be registered. This field is read only.
- **Business Name** is the name of the data structure in the UDDI registry. It is assumed that the business has already been registered in the UDDI. Choose the Business name from the list. This field is required.

Figure 9-6 Publish Service to UDDI Window from a UDDI Source



4. Click **OK** in the Publish Service to UDDI window.

   The system verifies that the service specified has a valid WSDL and that the UDDI registry has accepted the new entry or updated an existing one. If it is successful, a confirmation message displays and the service is published to the registry. Once the service is published in the UDDI, it becomes available to be registered to a source, as described in "Registering Web Services from a UDDI Source".

   Any errors during the operation will result in an error message.

   Note that you can only register the service to a source if it uses a unique WSDL.

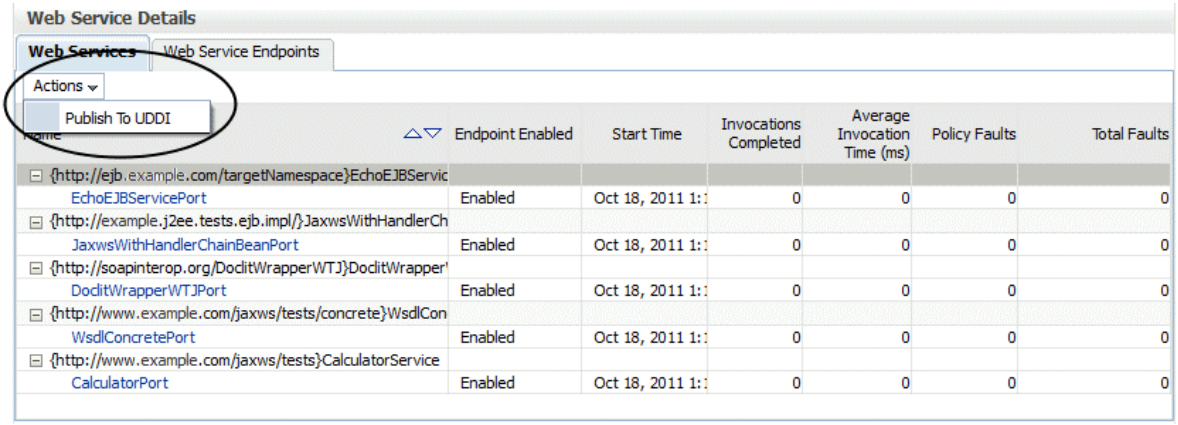
9.2.2 Publishing a Web Service to UDDI from an Application

You can publish web services to UDDI from the web services summary page for ADF and Java EE applications.

To publish a web service to UDDI from an application:

1. Navigate to the Web Services application summary page as described in "Viewing the Web Services Summary Page for an Application".
2. From the Web Service Details section of the page, select the **Web Services** tab, if it is not already selected, then select the service to be published.
3. Select **Actions**, then **Publish to UDDI.** See Figure 9-7.

Figure 9-7 Web Services Summary Page with Publish to UDDI Selected



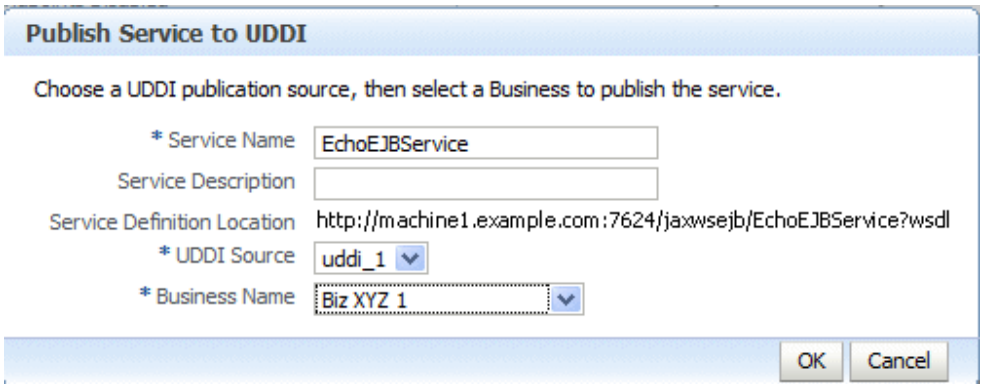
4. In the Publish Service to UDDI dialog box (Figure 9-8), enter the information about the service to be published:

   - **Service Name** is the name of the web service to be published to the UDDI registry. This field is required.
   - **Service Description** is a description of the selected web service.
   - **Service Definition Location** is prepopulated with the URL location of the service definition (This field is read-only.)
   - **UDDI Source** is a logical name for the UDDI registry source. Choose the UDDI source from the list. This field is required.

   > **Note:**
   >
   > The list contains the UDDI sources registered in the domain that have been enabled for publishing. For more information about registered sources, see "Introduction to Registering Web Services and Sources".

   - **Business Name** is the name of the data structure in the UDDI registry. It is assumed that the business has already been registered in the UDDI. Choose the business name from the list. This field is required.

Figure 9-8 Publish Service to UDDI Dialog Box



5. Click **OK** to connect to the external UDDI registry and register the web service.

   Upon successfully registering the service, a confirmation message displays. Any errors during the operation will result in an error message.

9.2.3 Configuring the Proxy Server for UDDI

To access URLs outside of your firewall, you must use a proxy to publish a service to UDDI.

Before starting Oracle WebLogic, you must set the Java system properties defined in Table 9-1. You can set them as environment variables, or in Oracle WebLogic startup files.

Table 9-1 Java System Properties Used to Specify the Proxy Server for UDDI

Property	Description
`proxySet=true`	Flag that specifies that the WebLogic proxy properties should be used.
`http.proxyHost=`*proxyHost*	Name of the host computer on which the proxy server is running.
`http.proxyPort=`*proxyPort*	Port to which the proxy server is listening.
`http.nonProxyHosts=`*hostname* | *hostname* |*...*	List of hosts that should be reached directly, bypassing the proxy. Separate each host name using a | character.

For example:

```
set PROXY_SETTINGS="-DproxySet=true -Dhttp.proxyHost=www-proxy.example.com -
Dhttp.proxyPort=80 -Dhttp.nonProxyHosts=localhost|${HOST}|*.example.com"
```

A

Web Service Audit Events Reference

You can use the web service and OWSM events that can be audited using the Oracle Fusion Middleware Audit Framework listed in this section as a reference.
For complete details on using the Oracle Fusion Middleware Audit Framework, see "Using Audit Analysis and Reporting" in *Securing Applications with Oracle Platform Security Services*.

This appendix contains the following sections:

- Web Service Audit Events
- Pre-built Audit Reports

A.1 Web Service Audit Events

This section describes the components that are audited and the types of events that can be audited.

- "OWSM-AGENT Events and Attributes"
- "OWSM-PM-EJB Events and Attributes"
- "Web Services Policy Attachment Events and Attributes"
- "Oracle Web Services Events and Attributes"

A.1.1 OWSM-AGENT Events and Attributes

The following table summarizes the OWSM-AGENT event types that can be audited.

Table A-1 OWSM-AGENT Events

Event Category	Event Type	Attributes used by Event
UserSession	Authentication	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Resource, AssertionName, CompositeName, Endpoint, AgentMode, ModelObjectName, Operation, ProcessingStage, Version, Protocol

Table A-1 (Cont.) OWSM-AGENT Events

Event Category	Event Type	Attributes used by Event
Authorization	Check Authorization	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Resource, AssertionName, CompositeName, Endpoint, AgentMode, ModelObjectName, Operation, ProcessingStage, Version, Protocol
PolicyEnforcement	Enforce Confidentiality	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Resource, AssertionName, CompositeName, Endpoint, AgentMode, ModelObjectName, Operation, ProcessingStage, Version, Protocol
—	Enforce Integrity	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Resource, AssertionName, CompositeName, Endpoint, AgentMode, ModelObjectName, Operation, ProcessingStage, Version, Protocol
—	Enforce Policy	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Resource, AssertionName, CompositeName, Endpoint, AgentMode, ModelObjectName, Operation, ProcessingStage, Version, Protocol

A.1.2 OWSM-PM-EJB Events and Attributes

The following table summarizes the OWSM-PM-EJB event types that can be audited.

Table A-2 OWSM-PM-EJB Events

Event Category	Event Type	Attributes used by Event
Assertion Template Authoring	Create Assertion Template	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version
—	Delete Assertion Template	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version, ToVersion
—	Modify Assertion Template	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version
Policy Authoring	Create Policy	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version
—	Delete Policy	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version, ToVersion,
—	Modify Policy	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version

Table A-2 (Cont.) OWSM-PM-EJB Events

Event Category	Event Type	Attributes used by Event
PolicySet Authoring	Create PolicySet	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version
—	Delete PolicySet	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version, ToVersion,
—	Modify PolicySet	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, Resource, Version

A.1.3 Web Services Policy Attachment Events and Attributes

The following table summarizes the WS-Policy attachment event types that can be audited.

Table A-3 WS-Policy Attachment Events

Event Category	Event Type	Attributes used by Event
WS-PolicyAttachment	Policy Attachment Event	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Target, Resource, PolicyChangeType, PolicyURI, PolicyCategory, PolicyStatus, ServiceEndPoint, PolicySubjRescPattern

A.1.4 Oracle Web Services Events and Attributes

The following table summarizes the Oracle Web Services event types that can be audited.

Table A-4 Oracle Web Services Events

Event Category	Event Type	Attributes used by Event
WS-Processing	Request Received	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Target, Resource, Protocol, Endpoint, Operation, FaultUrl
—	Response Sent	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Target, Resource, Protocol, Endpoint, Operation, FaultUri
WS-Fault	Soap Fault Event	ComponentType, InstanceId, HostId, HostNwaddr, ProcessId, OracleHome, HomeInstance, ECID, RID, ContextFields, SessionId, TargetComponentType, ApplicationName, EventType, EventCategory, EventStatus, TstzOriginating, ThreadId, ComponentName, Initiator, MessageText, FailureCode, RemoteIP, Target, Resource, URI, Source, Protocol, Endpoint, Operation

A.2 Pre-built Audit Reports

Oracle Fusion Middleware Audit Framework provides a range of out-of-the-box reports that are accessible through Oracle Business Intelligence Publisher.

The OWSM audit reports are organized as follows:

- User Activities
  - Authentication History
  - Authorization History
- Errors and Exceptions
  - All Errors and Exceptions
  - Authentication Failures
  - Authorization Failures
- All Events
- Policy Management
  - Assertion Template Management
  - Web Services Policy Management

- Policy Enforcements
  - Confidentiality Enforcements
  - Policy Enforcements
  - Message Integrity Enforcements
  - Violations
- Request Response
- Policy Attachments

OWSM Audit Reports

When your audit data resides in a database, you can run pre-defined Oracle Business Intelligence Publisher reports and create your own reports on the data.

For details about "configuring, generating, and managing the audit reports", see *Administrator's Guide for Oracle Business Intelligence Publisher*.

Index

A