Oracle Enterprise Gateway 11*g*: Security Management for SOA and Cloud

Activity Guide

D73680GC10
Edition 1.0
May 2012
Dxxxx

ORACLE

Author

Iris Li

Technical Contributors and Reviewers

Gary Barg, Patrice Goutin, Kenneth Heung, Sidharth Mishra

Editors

Arijit Ghosh, Smita Kommini, Rashmi Rajagopal, Richard Wallis

Graphic Designers

Satish Bettegowda, Seema Bopaiah

Publishers

Jayanthy Keshavamurthy, Sumesh Koshy

This book was published using: oracle*tutor*

Table of Contents

Practices for Lesson 1: Introduction

Chapter 1

Practices for Lesson 1: Overview

Practices Overview

There are no practices for this lesson titled “Oracle Enterprise Gateway 11*g*: Security Management and Control for SOA and the Cloud – Course Introduction”.

General Notes

There are no notes.

Practices for Lesson 2: Web Services Security Overview

Chapter 2

Practices for Lesson 2: Overview

Practices Overview

There are no practices for this lesson titled “Oracle Enterprise Gateway 11*g*: Security Management and Control for SOA and the Cloud – Web Services Security Overview”.

General Notes

There are no notes.

Practices for Lesson 3: Getting Started with Oracle Enterprise Gateway 11g

Chapter 3

Practices for Lesson 3: Overview

Practices Overview

In these practices, you will make yourself familiar with the configuration of the Enterprise Gateway instance and the user interface of Policy Studio.

Practice 3-1: Explore the Enterprise Gateway Configuration

Overview

In this practice, you start Enterprise Gateway and Policy Studio, connect Policy Studio to the Gateway, and explore the configuration of the running Gateway instance.

Tasks

Your tasks here are to:

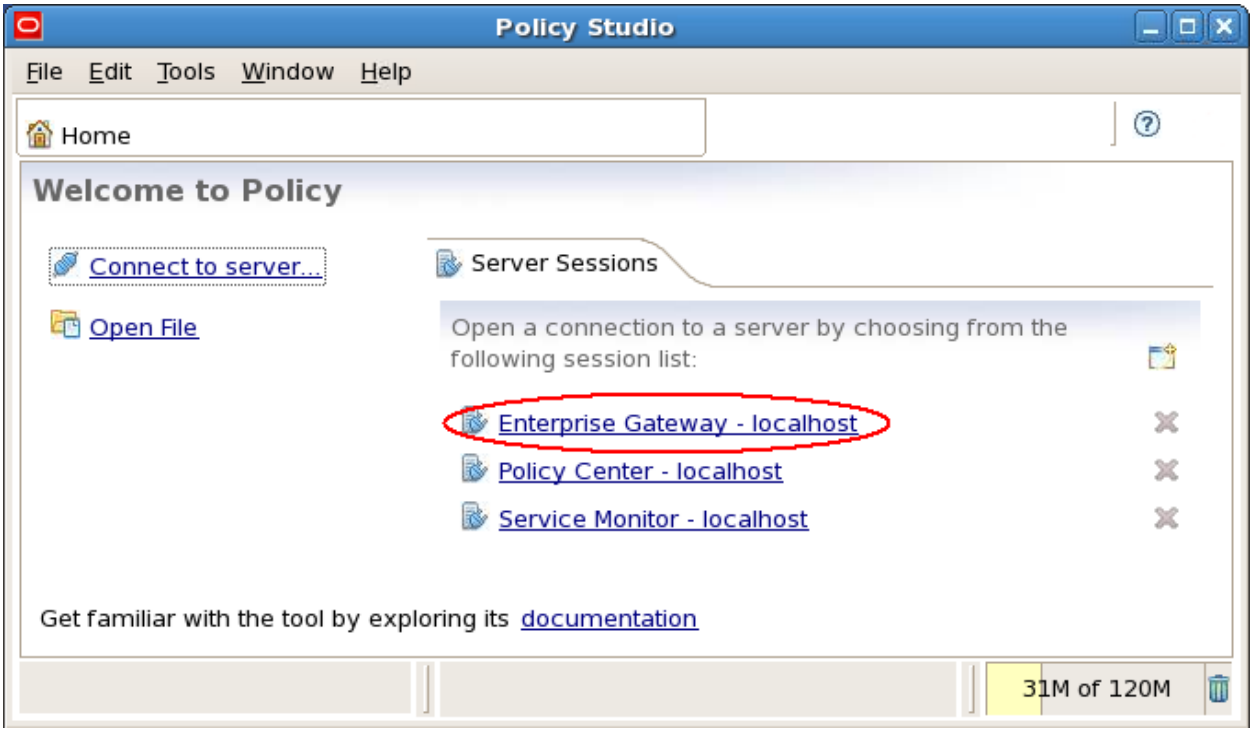
- Start Enterprise Gateway
- Start Policy Studio
- Connect Policy Studio to Enterprise Gateway
- Explore the active configuration of the Enterprise Gateway instance

1. Start the OEG gateway:
   a. Locate the “Start Enterprise Gateway” launcher on the Desktop and double-click it.
   b. In the terminal window, wait until you see a message similar to:

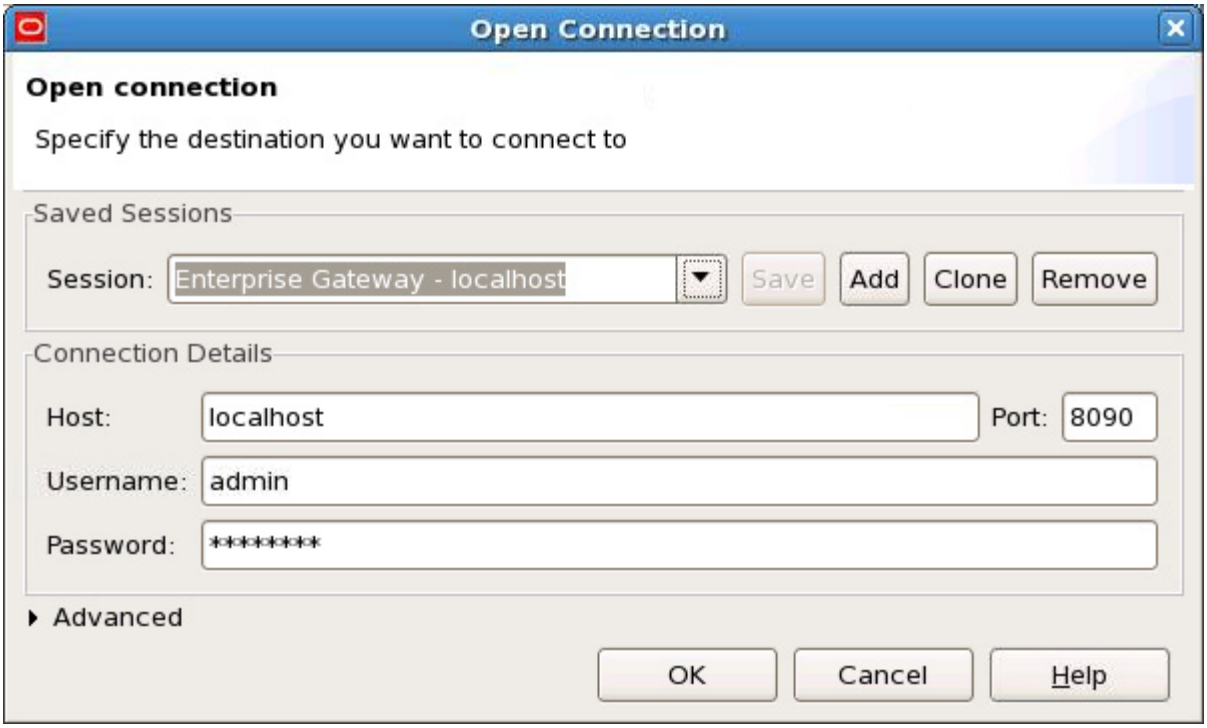
```
...
INFO  06/Jan/2012:10:30:39.466 [c16b78f0]  TCP interface
INFO  06/Jan/2012:10:30:39.466 [c16b78f0]  checking invariants for
interface *:8080
INFO  06/Jan/2012:10:30:39.466 [c16b78f0]  listen on address
0.0.0.0/8080
INFO  06/Jan/2012:10:30:38.573 [c16b78f0]  TCP interface
INFO  06/Jan/2012:10:30:38.573 [c16b78f0]  checking invariants for
interface *:8090
INFO  06/Jan/2012:10:30:38.573 [c16b78f0]  listen on address
0.0.0.0/8090
INFO  06/Jan/2012:10:30:38.838 [c16b78f0]  operations DB responder
waiting for requests on /ops/
...
INFO  06/Jan/2012:10:30:39.627 [c16b78f0]  starting 4 idle netsvc
threadpool threads. Max 4096
INFO  06/Jan/2012:10:30:39.628 [c16b78f0]  service started (version
6.3.0-2011-12-20, pid 10869)
INFO  06/Jan/2012:10:30:39.633 [41ee9940]  loaded netservice library
```

   c. Minimize the “Start Enterprise Gateway” terminal window.
2. Start Policy Studio from the desktop shortcut “Start Policy Studio”.

3. On the Home tab, connect to the OEG Gateway by clicking “Enterprise Gateway – localhost” under Server Sessions.

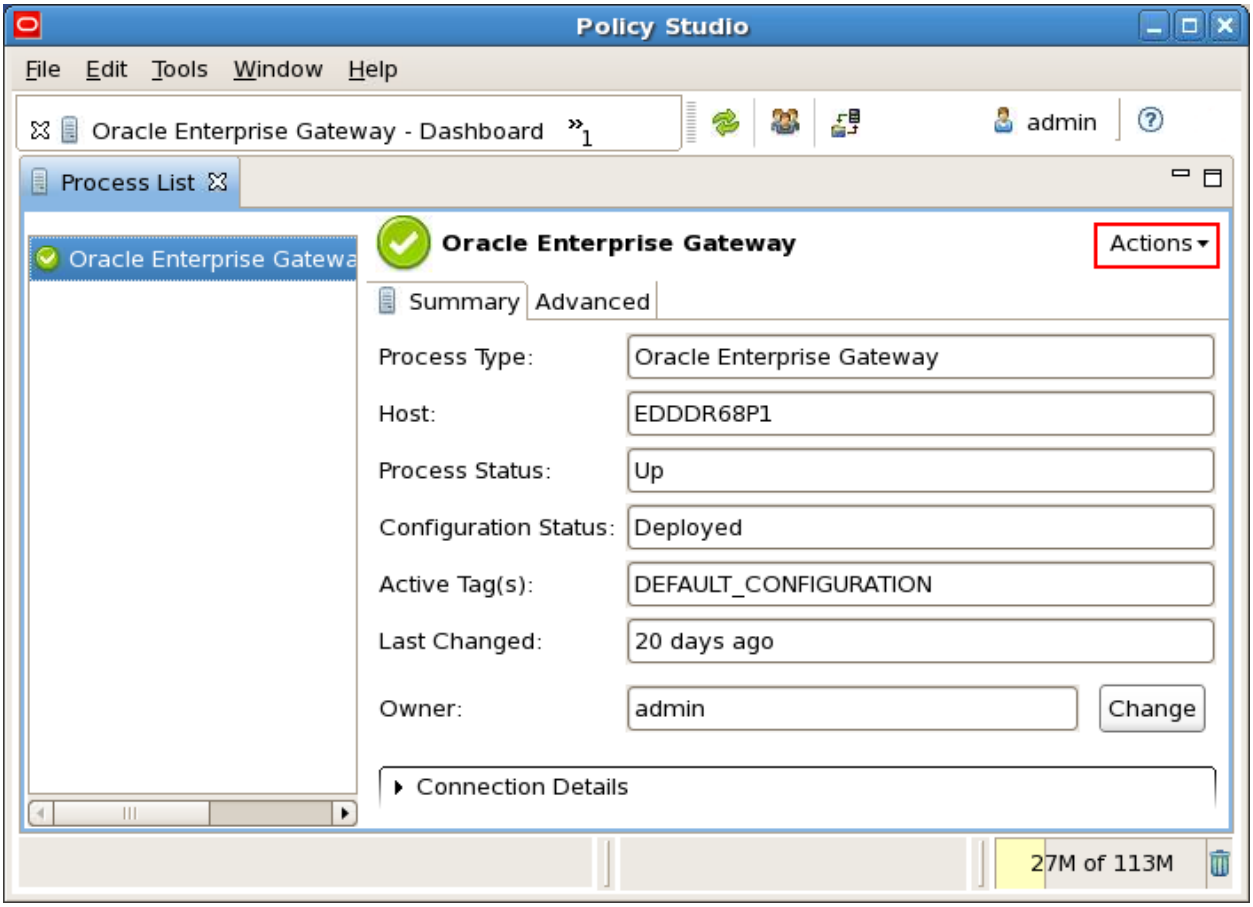


4. In the Open Connection window, accept the default settings, and click OK.



When the connection to the server has been made, the Oracle Enterprise Gateway Dashboard tab is displayed. You should see only one process currently managed by Policy Studio.

5. To edit the current active configuration for a process, click Actions, and select “Edit Active Configuration” from the drop-down list.



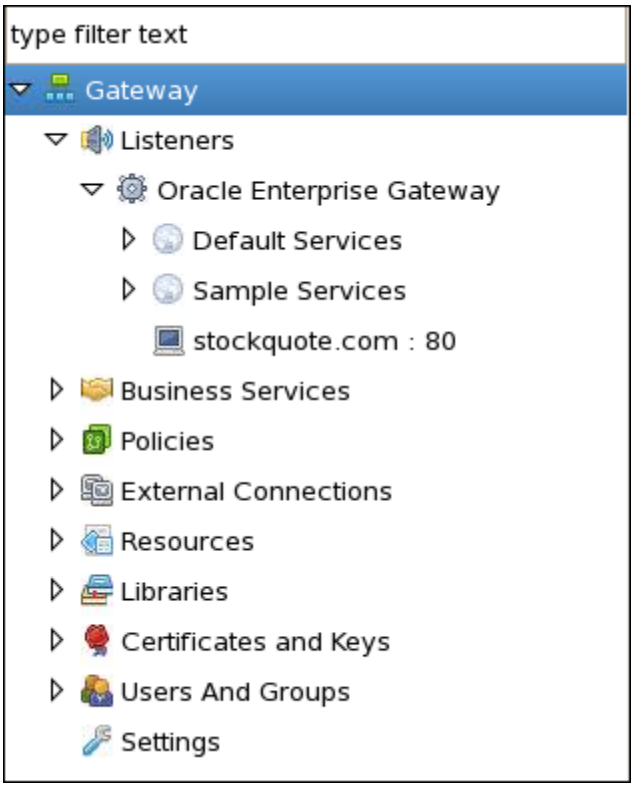
When the Enter Passphrase dialog box is displayed, prompting for the password, just press OK (you don’t need to enter anything here).

Note: All sensitive server configuration data (password, keys, and so on) can be encrypted by using a passphrase. If you wish to do this, enter a password in the passphrase key field when connecting. You must use this password thereafter when connecting to the server.

The active server process configuration is loaded and displayed in a new page named in the following format: *processname* [*hostname*]. In this example, it is Oracle Enterprise Gateway [EDDR68P1].

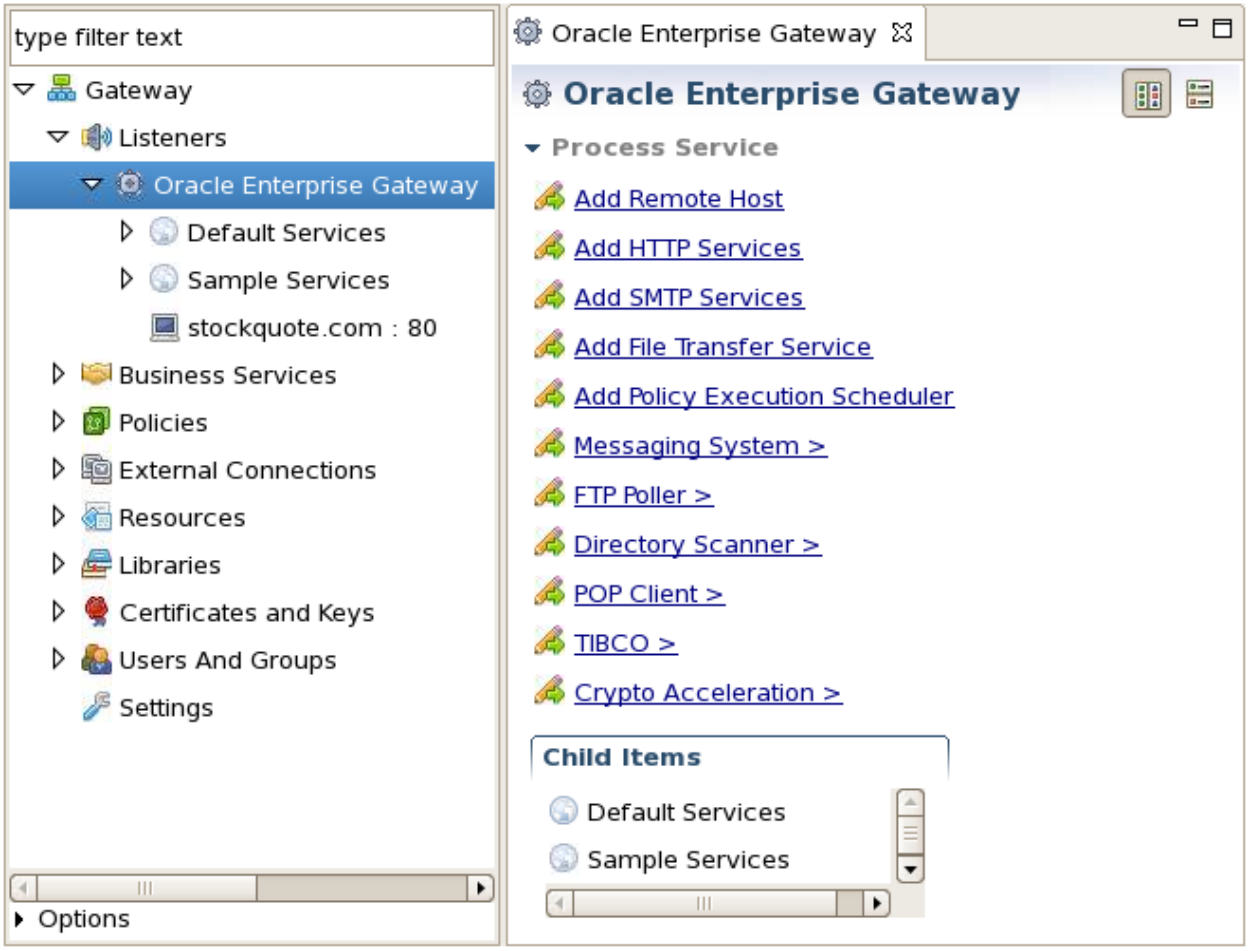
Explore Listeners

1. In the left navigation pane, expand the Gateway > Listeners > Oracle Enterprise Gateway node. The node contains two preconfigured HTTP Services groups: Default Services and Sample Services, and a remote host definition, which are shipped with Enterprise Gateway.

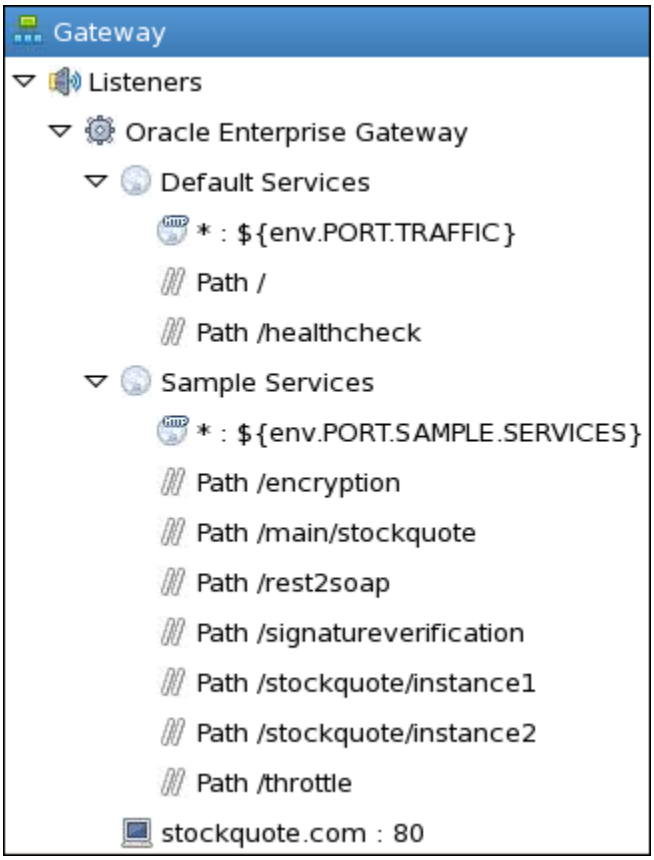


2. Click the Oracle Enterprise Gateway node, view the listeners and services you can add at the process level displayed in the right pane.

   Note: Alternatively, you can right-click the process to access the same configurable features in a drop-down menu.



3. Expand the Default Services and Sample Services nodes:

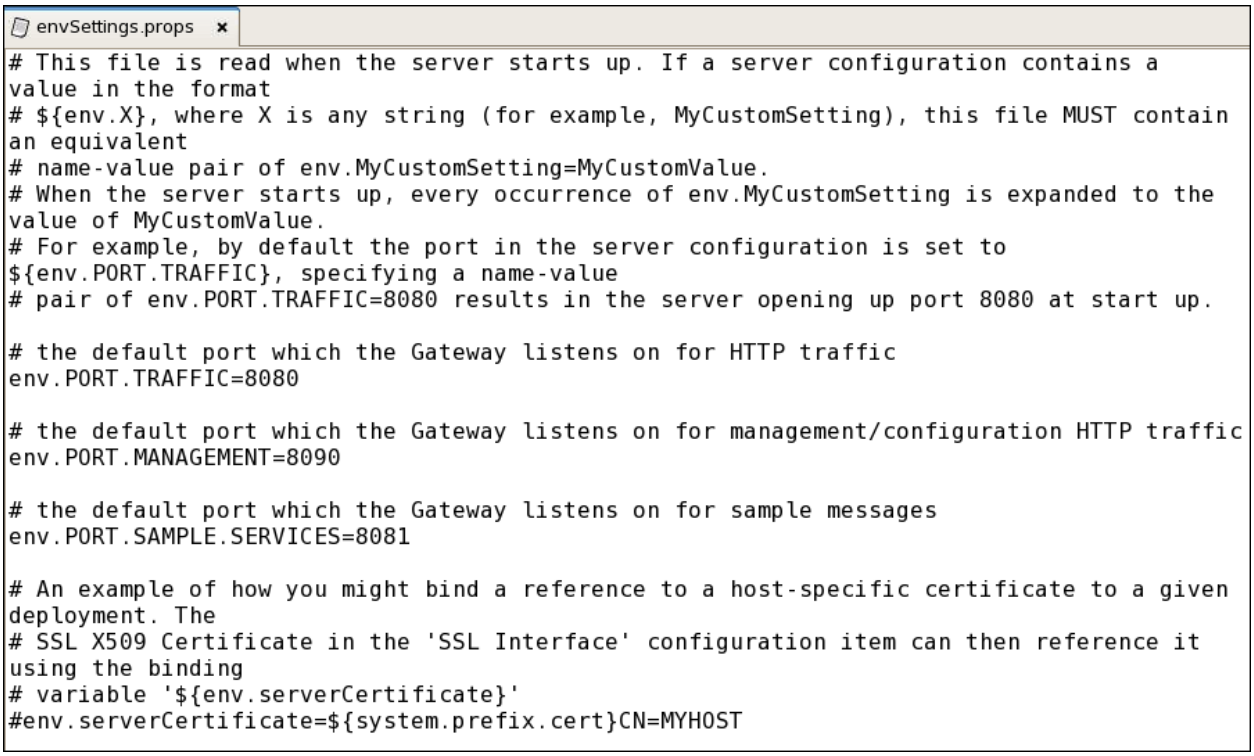


Review the content. You should see:

- The Default Services group uses an HTTP interface running on the port specified in the ${env.PORT.TRAFFIC} environment variable.
- The Sample Services group uses an HTTP interface running on the port specified in the ${env.PORT.SAMPLE.SERVICES} environment variable.
- Several defined relative paths

4. To view the default ports that the Gateway listens on, perform the following steps:
   a. Open a Text Editor.
   b. Click File menu, select Open…

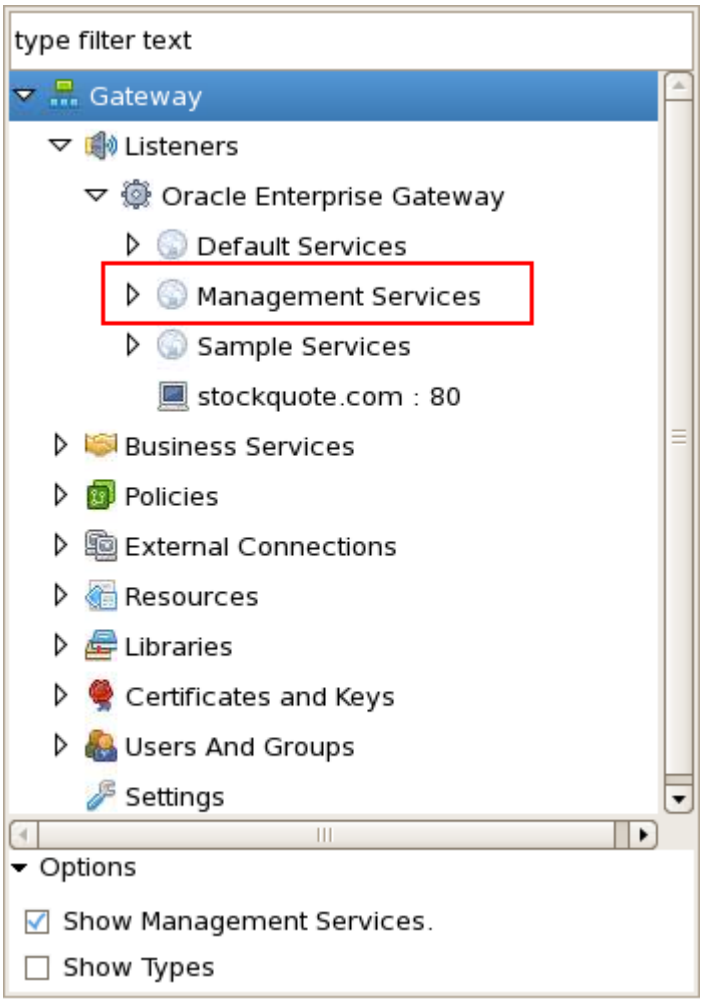
c. In the Open Files… window, navigate to the `/u01/app/oracle/oeg11g/enterprisegateway/conf` folder, open the `envSettings.props` file. You should see content resembling the following screenshot:



If you want to use different ports, you can make the changes here, save the file, and restart Enterprise Gateway to make the changes effective.

d. Close the Text Editor.

e. To make the Management Services group visible in Policy Studio, expand the Options link at the bottom of the navigation tree, and select the Show Management Services option. You should see the Management Services node displayed in the navigation tree under the Oracle Enterprise Gateway process.

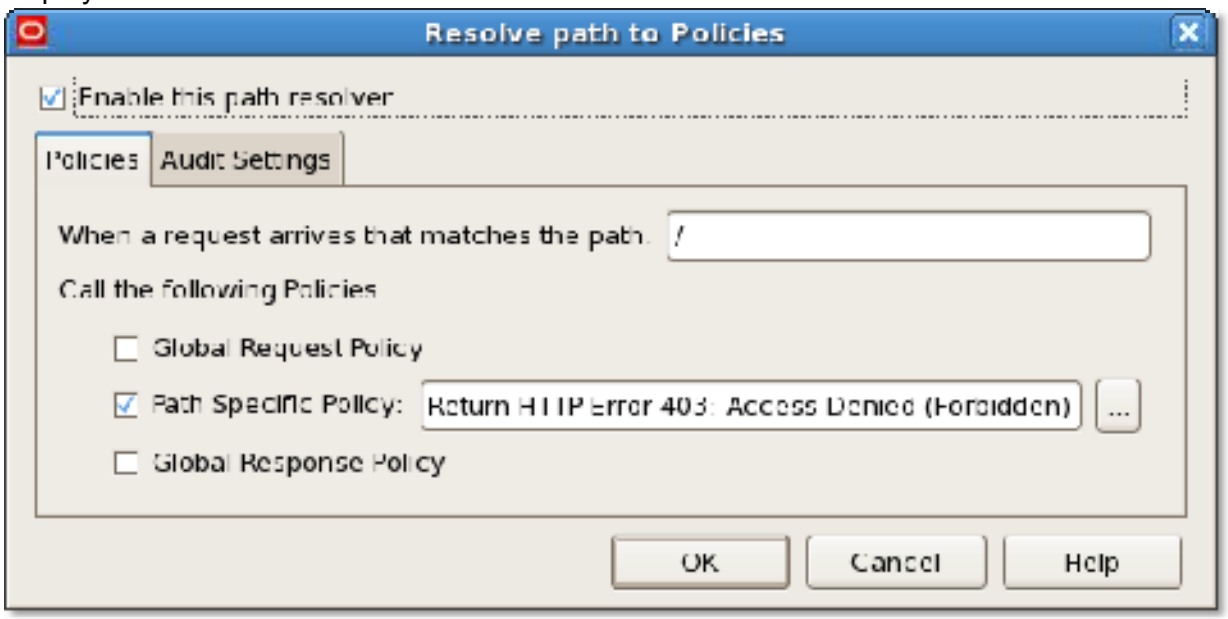


f. You don't need to edit the Management Services in this course, so deselect the Show Management Services option to hide it.

Note that the navigation frame is refreshed every time you enable or disable the Show Management Services option.

5. To view the mapping policy of the relative path, perform the following steps:
   a. Expand the Gateway > Listeners > Oracle Enterprise Gateway > Default Services node.
   b. Right-click the `// Path /` node.

c. Select Edit from the context menu. The Resolve path to the Policies window is displayed as shown below:

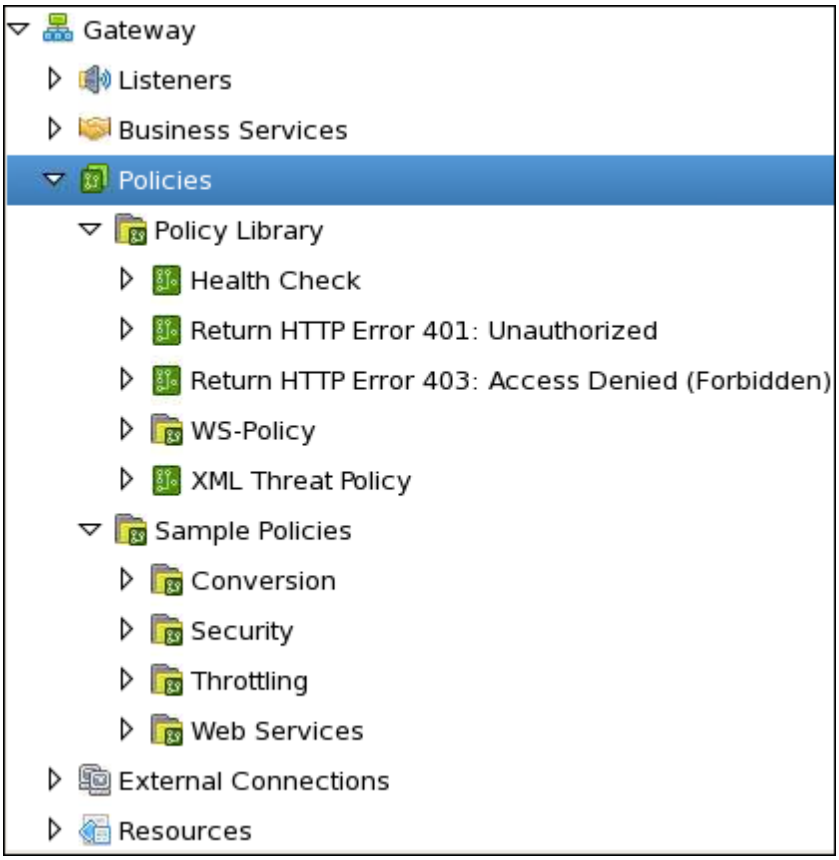


This means any requests sent to http://localhost:8080/ will be returned with an Access Denied error.

d. To verify this, open the Firefox browser from the desktop, and type the above URL in the address bar. You should see Access Denied message displayed.

Explore Policies

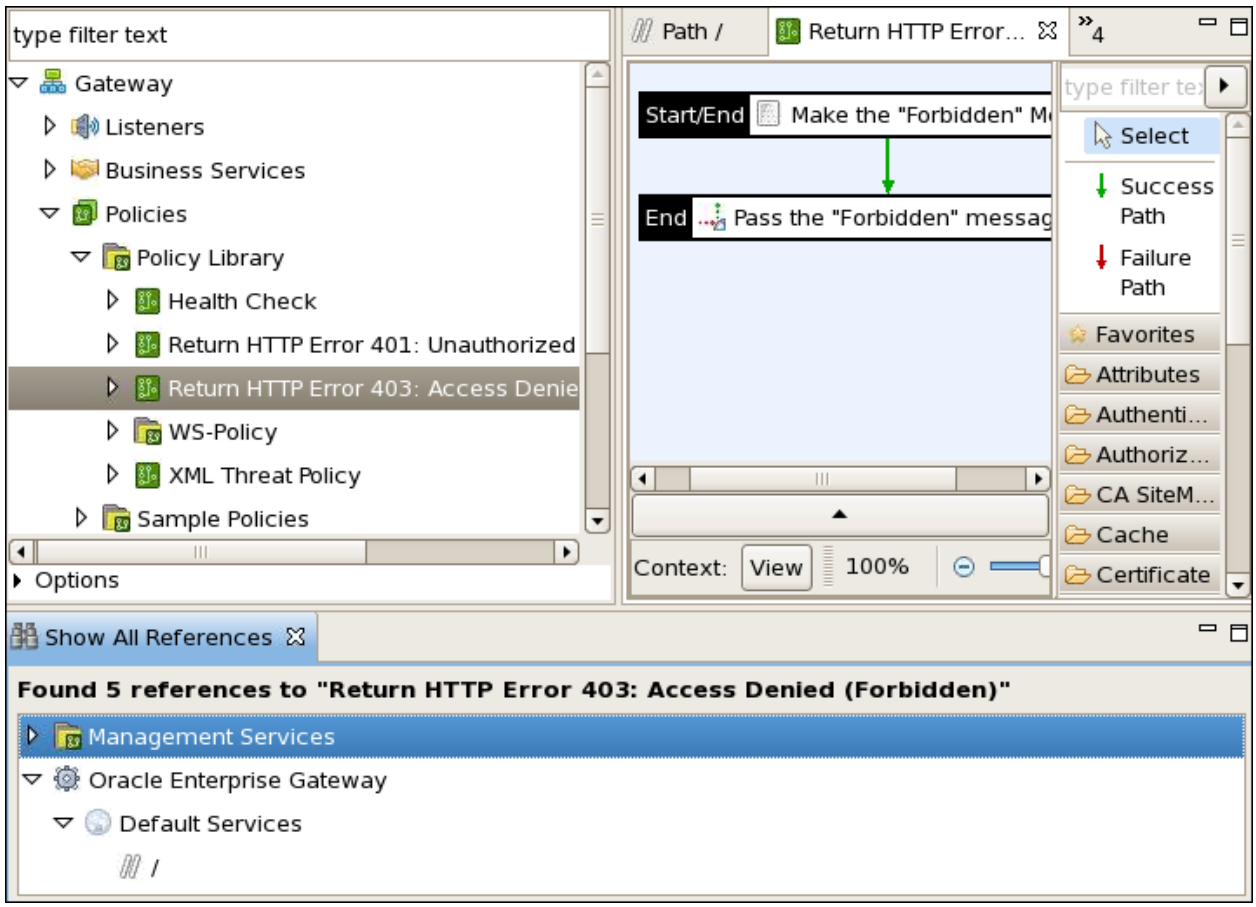
1. In the left navigation pane, expand the Gateway > Policies node. Under this node, you should see two policy containers: Policy Library and Sample Policies, resembling the following screenshot:



   The two containers store some general purpose policies, and some sample policies that ship with Enterprise Gateway.

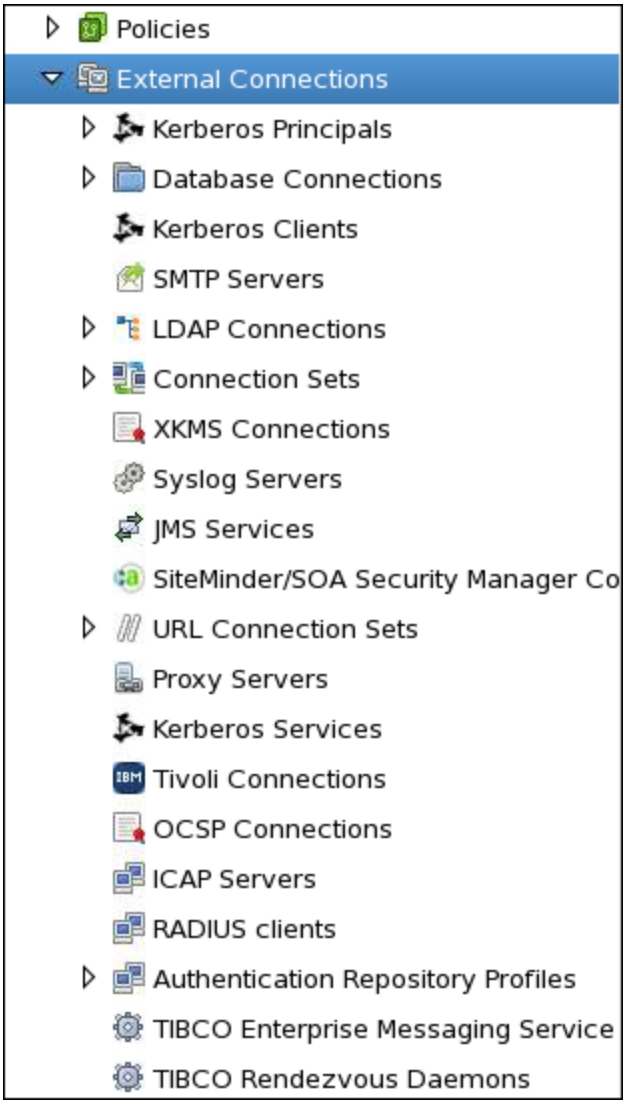
2. To view what relative paths or web services use the policy, for example “Return HTTP Error 403: Access Denied (Forbidden)”, perform the following steps:
   a. Right-click the “Return HTTP Error 403: Access Denied (Forbidden)” policy.
   b. Select Show All References from the context menu.

c. All the references to this policy are displayed at the bottom of Policy Studio as shown in the screenshot below:

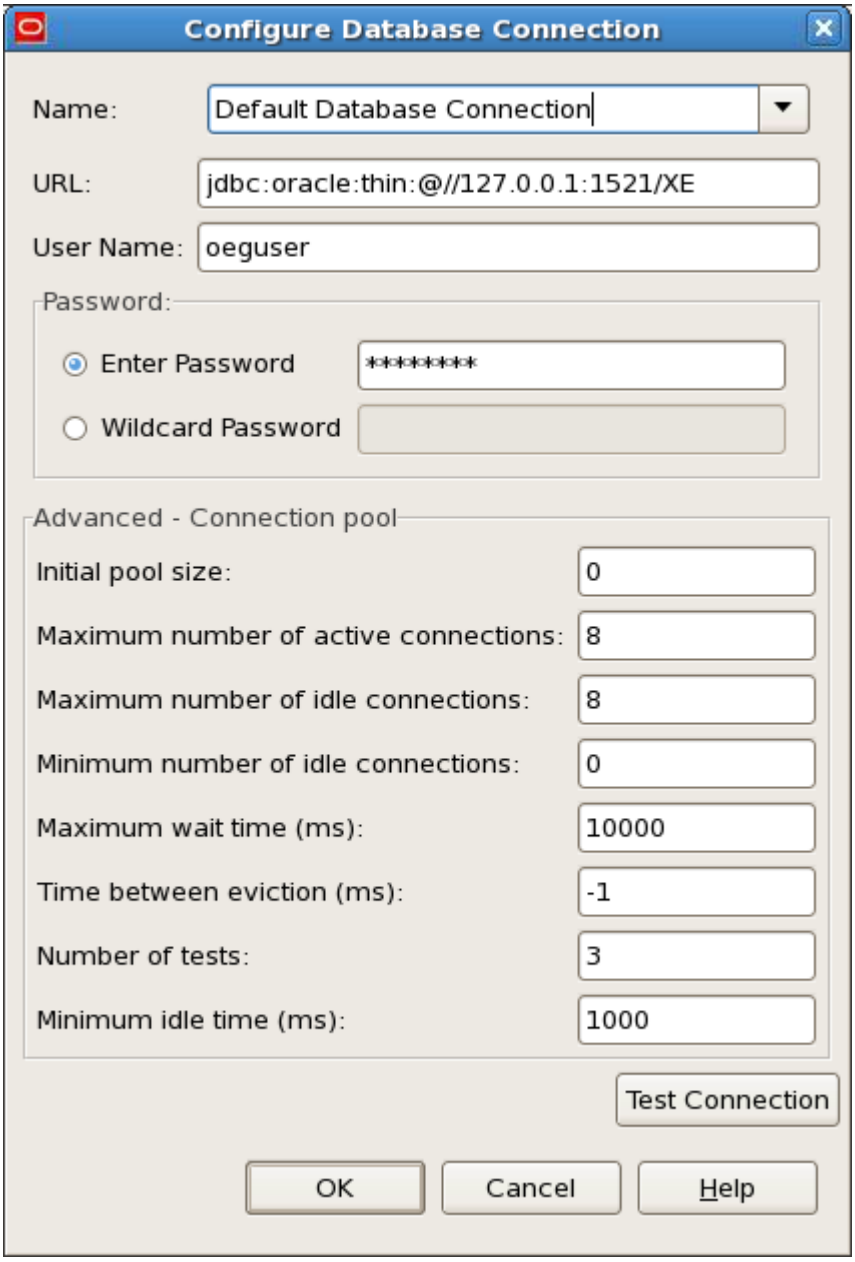


Explore External Connections

1. In the left navigation pane, expand the Gateway > External Connections node. You should see a list of configurable external connections displayed as shown in the screenshot below:



2. Expand the Database Connections folder, right-click Default Database Connection, and select Edit from the pop-up menu. The Configure Database Connection dialog box is displayed, as shown in the screenshot below:



In this course, the default database that the Gateway connects to is Oracle XE. The username and password to access the database are `oeguser` and `welcome1`.

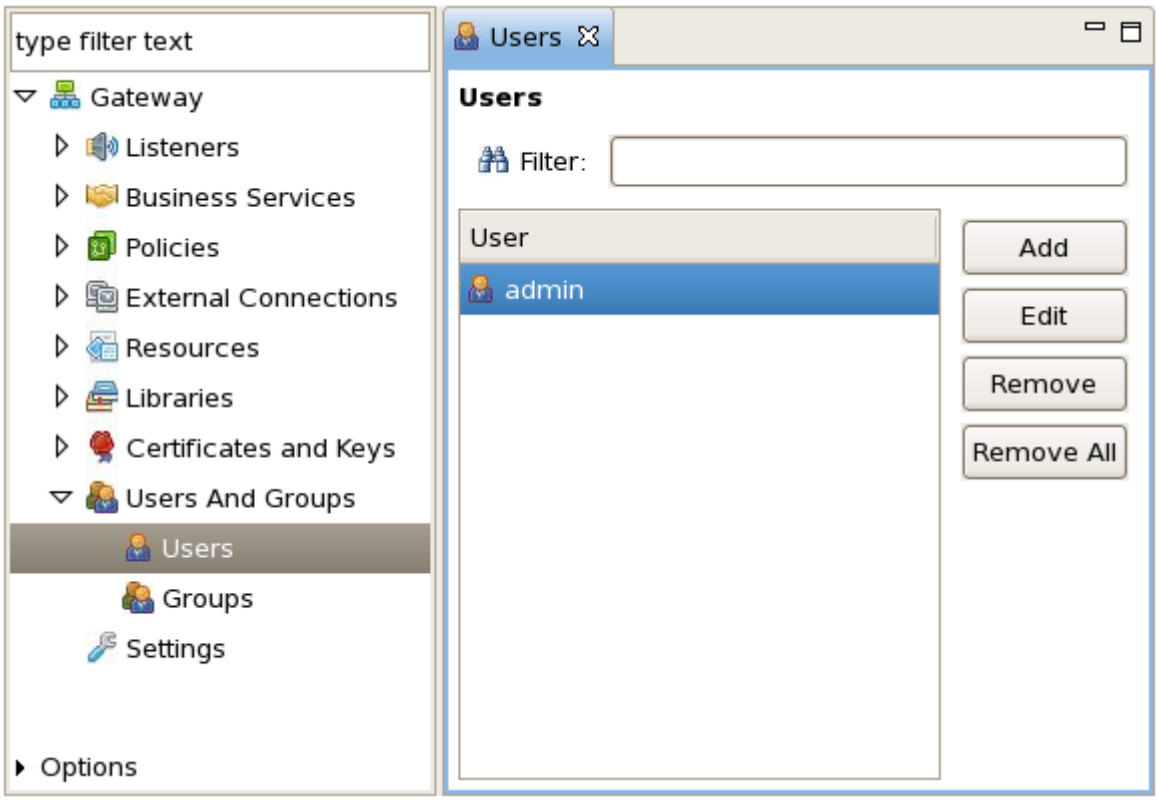
3. Click the Test Connection button to verify that the connection to the database is configured successfully. You should see a Connection test OK message. Click OK to close the Test Connection dialog box.
4. In the Configure Database Connection dialog box, click OK or Cancel to close this dialog box.

Explore Certificates and Private Keys

1. In the left navigation pane, select Certificates and Keys > Certificates to view the lists of certificates and private keys stored in the Certificate Store. Click the three tabs in the Certificates screen on the right:
   a. **Certificates with Keys**: You should see the server certificates with associated private keys.
   b. **Certificates**: A list of server certificates without any associated private keys should appear. The list is empty now.
   c. **CA**: A list of Certification Authority certificates with associated public keys appears.

Explore Users and Groups

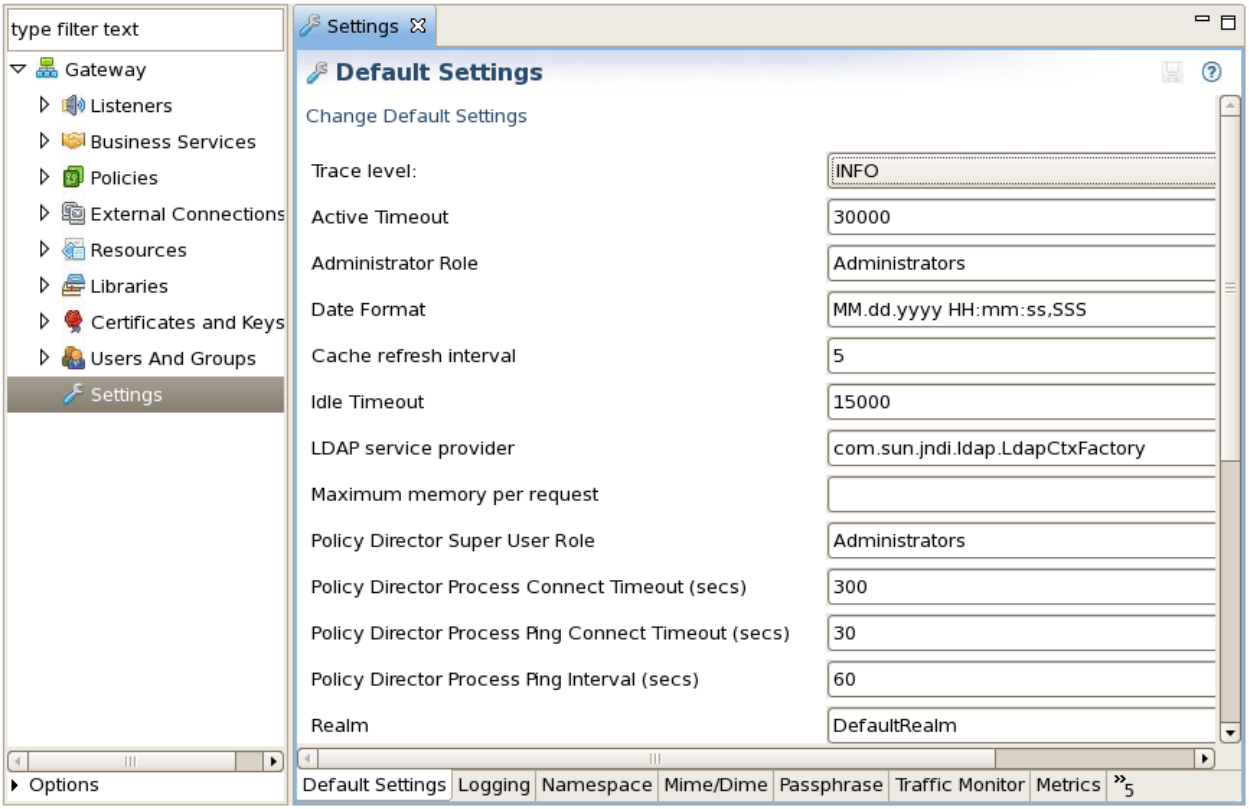
1. In the left navigation pane, select User And Groups > Users. You should see the Enterprise Gateway User Store contains only one user, `admin`, which comes with the OEG installation.



   a. Select `admin`, and click Edit. The Edit User dialog box is displayed. You can see that the username can't be edited, but you can change the default password, `changeme` here.
   b. Click Cancel to close the dialog box.

View Settings for Enterprise Gateway

1. In the left navigation pane, select Settings. The Default Settings tab is displayed in the right pane. In this screen, you can set several global configuration settings to optimize the behavior of Enterprise Gateway.



2. You can view other settings, such as Logging, Namespace, Mime/Dime, Passphrase, and so on, by clicking the tabs at the bottom of the right pane.
3. When you finish, leave Policy Studio open; you will use it in the next practice.

Practices for Lesson 4: Registering Web Services in OEG

Chapter 4

Practices for Lesson 4: Overview

Practices Overview

The goal of the practices in this lesson is to learn how to virtualize a back-end web service by registering it into Enterprise Gateway.

Practice 4-1: Registering the Web Service

Overview

A sample web service is predeployed to the WebLogic Server. In this practice, you will register this web service to Enterprise Gateway by importing its WSDL definition into the gateway, and deploy the virtual service to Enterprise Gateway.

Assumptions

The Enterprise Gateway and WebLogic Admin server are up and running. The sample web service (validateCC) has already been deployed on the Weblogic Administration server.

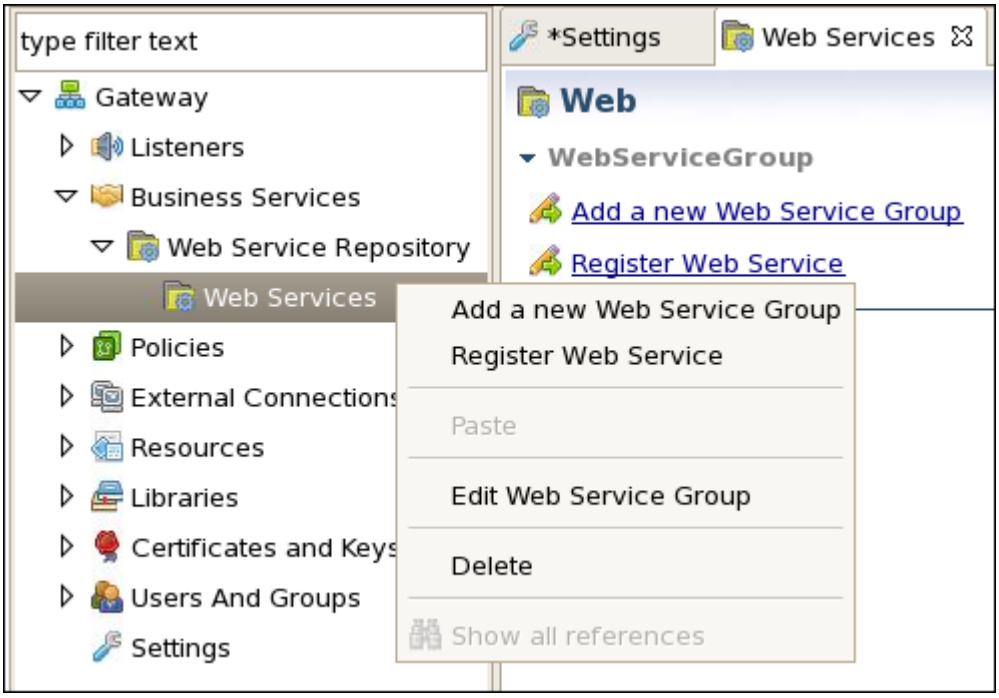
Tasks

The main tasks here are to:

- Start the WebLogic Administration server
- View the WSDL of the deployed web service in the browser
- Register the web service in Policy Studio
- Deploy the web service to the Gateway
- View the generated policy

1. To start the WebLogic Administration server, perform the following steps:
   a. Locate the “Start WebLogic Admin Server” icon on the Desktop, and double-click it.
   b. In the “Start WebLogic Admin Server” terminal window, wait until you see a message similar to:
      ```
      <Nov 3, 2011 2:00:51 PM UTC> <Notice> <WebLogicServer> <BEA-
      000360> <Server started in RUNNING mode>
      ```
   c. Minimize the “Start WebLogic Admin Server” terminal window.
2. The sample service you are using is a credit card validation service. To view the WSDL of the deployed sample web service, enter the following URL in the address field of the Firefox browser:
   ```
   http://localhost:7001/validatecc/ValidateCCPort?WSDL
   ```
   Note: In your lab environment, because no separate managed server is created for application deployment, the sample application is deployed to the Administration server.
3. Examine the WSDL file, and answer the following questions:
   Q: What type of interface does the web service have?
   A: The web service has a SOAP interface.
   Q: What operation does the web service offer?
   A: The operation is called validateCard.
4. Open Policy Studio if it’s not already open, and register the service. Web Services definitions can be imported from a file, a URL, or any UDDI registry. In this practice, you import the credit card validation service definition from the URL you just viewed.

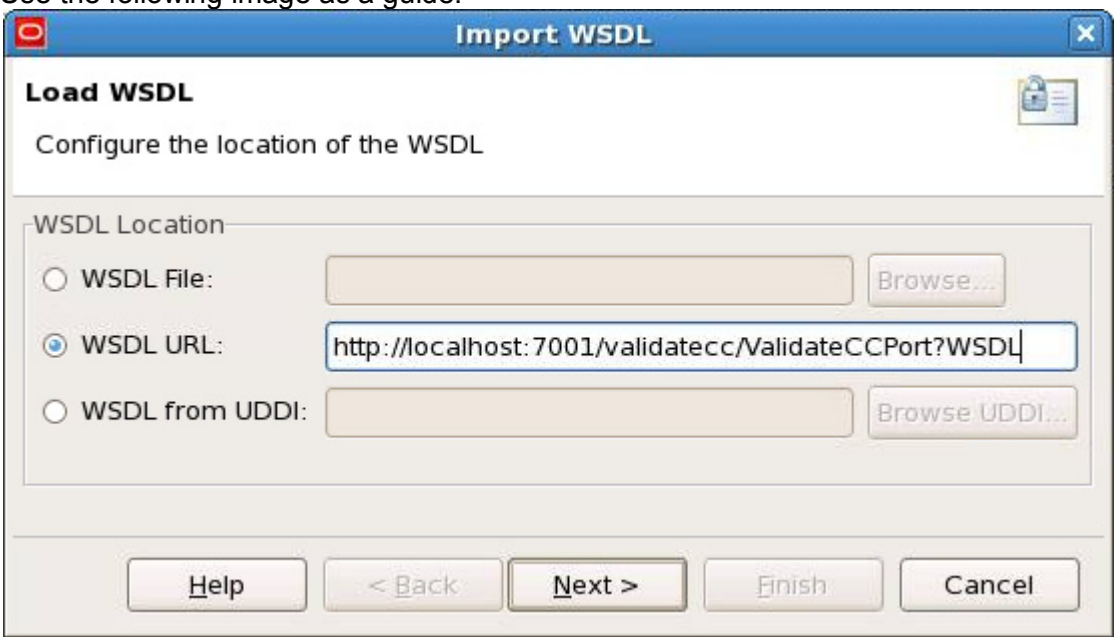
In the left pane, expand the Business Services > Web Service Repository tree node, right-click Web Services, and select Register Web Service from the drop-down menu. The WSDL Import wizard opens.



a. In the Import WSDL wizard, on the Load WSDL screen, select the WSDL URL option for WSDL Location, and use the following URL, which you can copy and paste from the web service opened earlier in Firefox:

```
http://localhost:7001/validatecc/ValidateCCPort?WSDL
```

Use the following image as a guide:



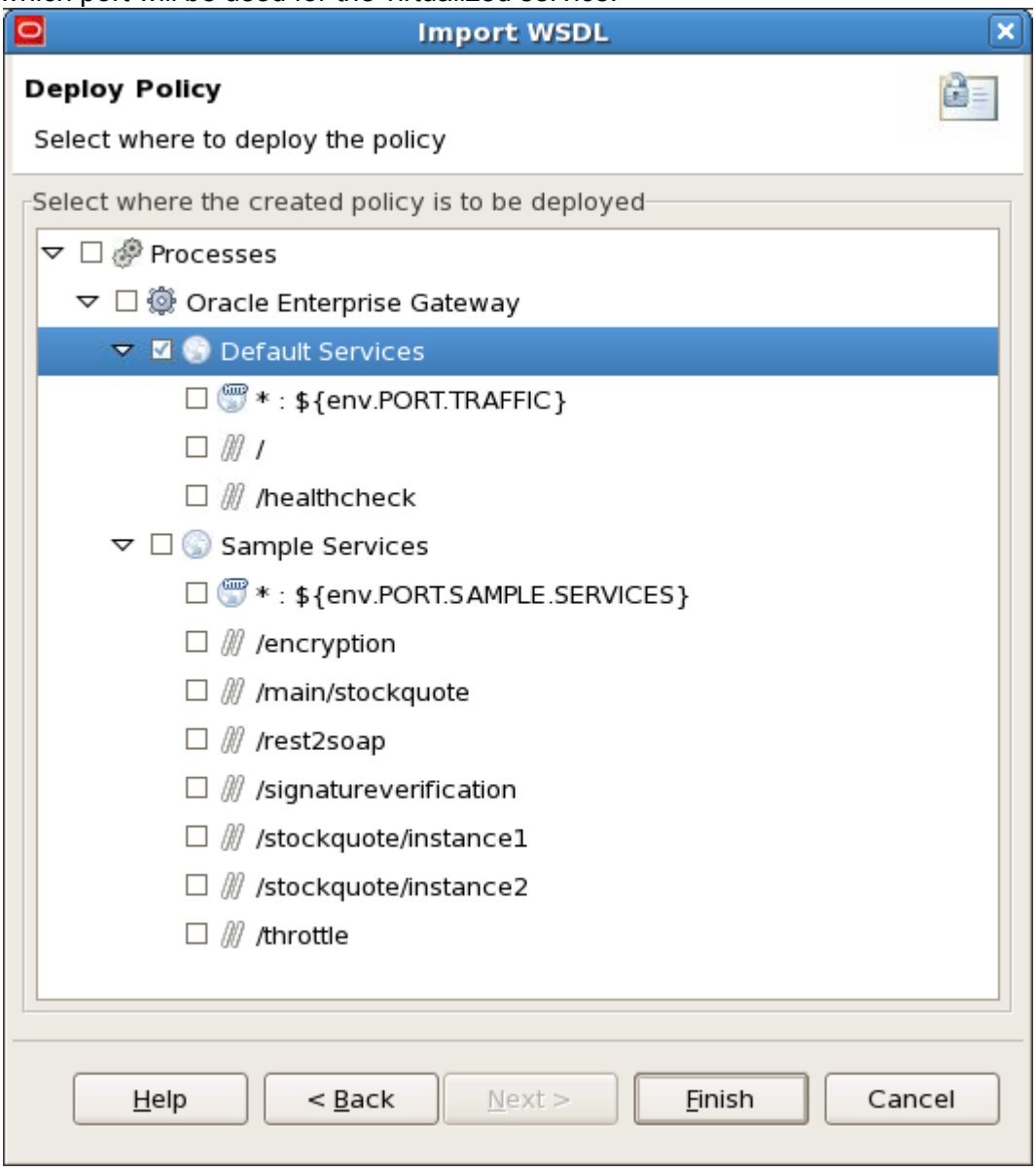
Click Next.

b. On the WSDL Operation screen, check the validateCard operation, and click Next. This is the operation you will virtualize in the Gateway. Use the following image as a guide:

c. For now, you will call the service without security, so do not select the “Secure this virtual service with a WS-Policy” check box when the WS-Policy Options screen appears. Click Next.

Note: This is where presupplied policies can be applied to the virtualized service. You can always go back to this policy configuration wizard by choosing “Configure Recipient WS-Policy” when right-clicking the service at any point.

d. On the Deploy Policy screen, select “Default Services” for deployment. This defines which port will be used for the virtualized service.



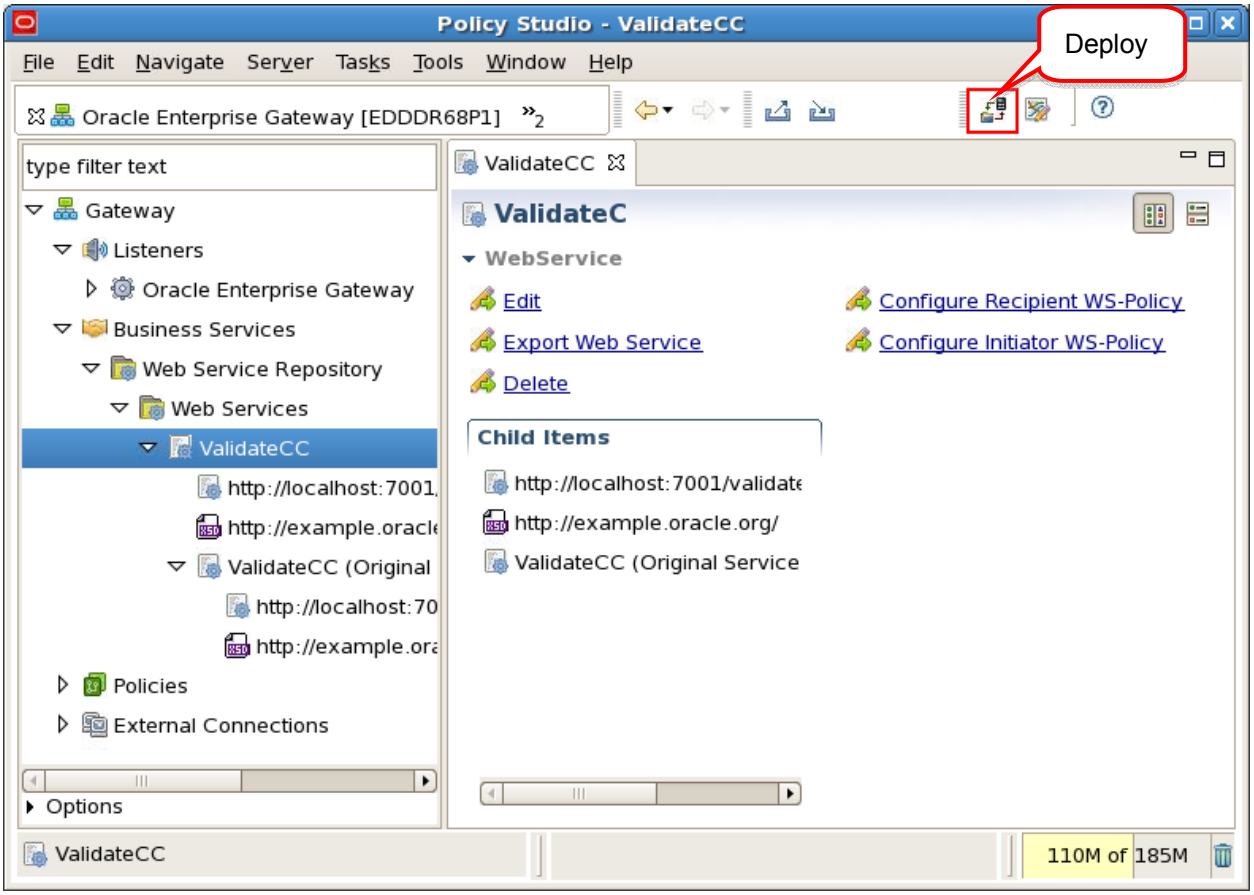
e. Click Finish.

f. A summary window appears, where you could overwrite some of the default behavior, such as routing or schema validation. Click OK to accept the defaults. The service is now virtualized at the Gateway level.

The following artifacts will be created:

- A web service definition inside the Web Services repository
- A web service context entry (under Listeners > Oracle Enterprise Gateway > Default Services)
- A default policy (under Policies > Generated Policies)
- A remote host representing the server that hosts the services

5. To deploy the service to the Gateway, click the Deploy icon on the Policy Studio toolbar or, alternatively, press F6.



The new configuration is now active.

The gateway is now virtualizing the validateCard operation of the sample service. Rather than connecting to the service directly, clients go through the Gateway. The Gateway then applies policies to the traffic to the destination web service.

6. To view the generated policy:
   a. Navigate to the policy generated for the ValidateCC under Policies > Generated Policies > Web Services.ValidateCC
   b. When you import WSDL files into the repository, this auto-generates a Service Handler that is used to control and validate requests to the web service and responses from the web service. Right-click the Service Handler for the ValidateCC filter, and select Edit from the context menu.
   c. In the “Configure the connection to the back-end Web Service” window, go to the WSDL tab. Notice that “Allow the Gateway to publish WSDL to clients” is selected. This option must be enabled if you want clients to send SOAP messages through the Gateway to access the service.
   d. Click Cancel to close the window.

e. To see the WSDL exposed to clients, point Firefox to:

`http://localhost:8080/validatecc/ValidateCCPort?WSDL` where:

- `8080` is the listing port of the default HTTP services where the gateway is listening for requests
- `validatecc/ValidateCCPort` is the URI of the service you are calling. This URI is the same as the original web service, and could be changed.

Practice 4-2: Testing the Web Service

Overview

In this practice, you will test the virtualized web service in Service Explorer by sending a message through Enterprise Gateway.

Tasks

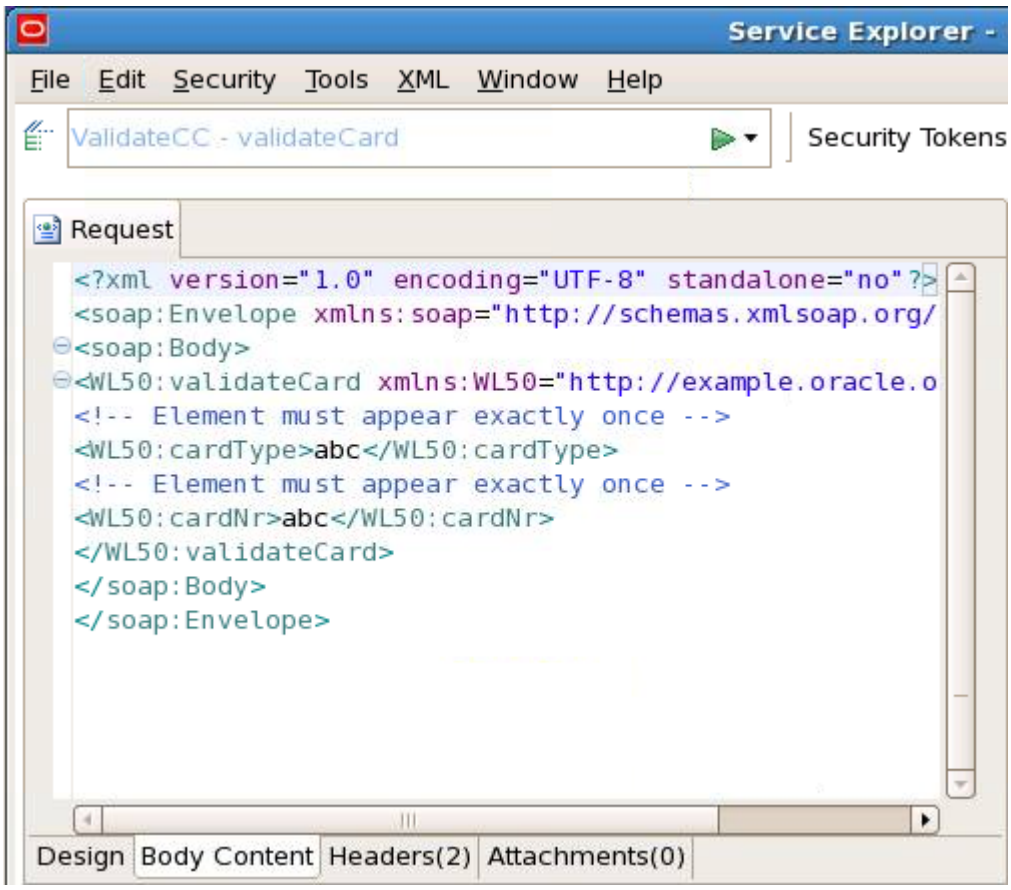
The main task here is to generate a SOAP request message to test the virtualized web service by using Service Explorer.

1. Start Service Explorer from the desktop shortcut.
2. To generate the test SOAP message from the WSDL file of the web service, perform the following steps:
   a. In Service Explorer window, click Import WSDL in the Service Explorer toolbar to launch the Load WSDL wizard.
   b. In the Load WSDL screen, select the WSDL URL option, and enter the URL of the WSDL exposed to clients:

      ```
      http://localhost:7001/validatecc/ValidateCCPort?WSDL
      ```

      Click Next.
   c. In the WSDL Operations screen, select the validateCard operation, and click Finish.

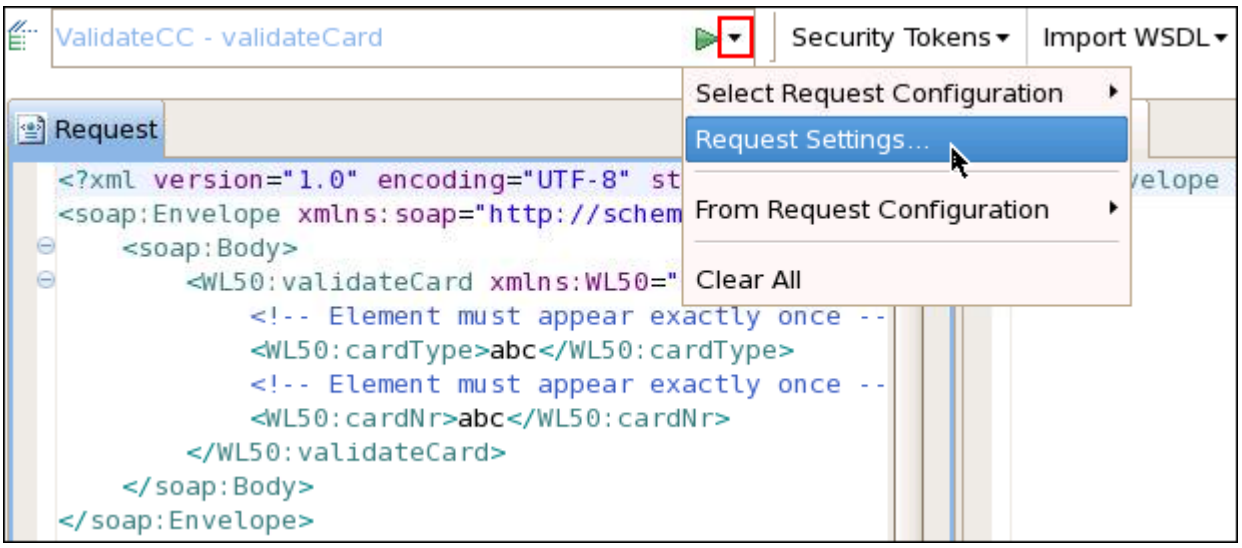
d. In the Request tab on the left, you should see the panel is automatically populated with the SOAP message, which resembles the following screenshot:



Note: The values of `cardType` and `cardNr` in your message might not be the same as shown in the screenshot.

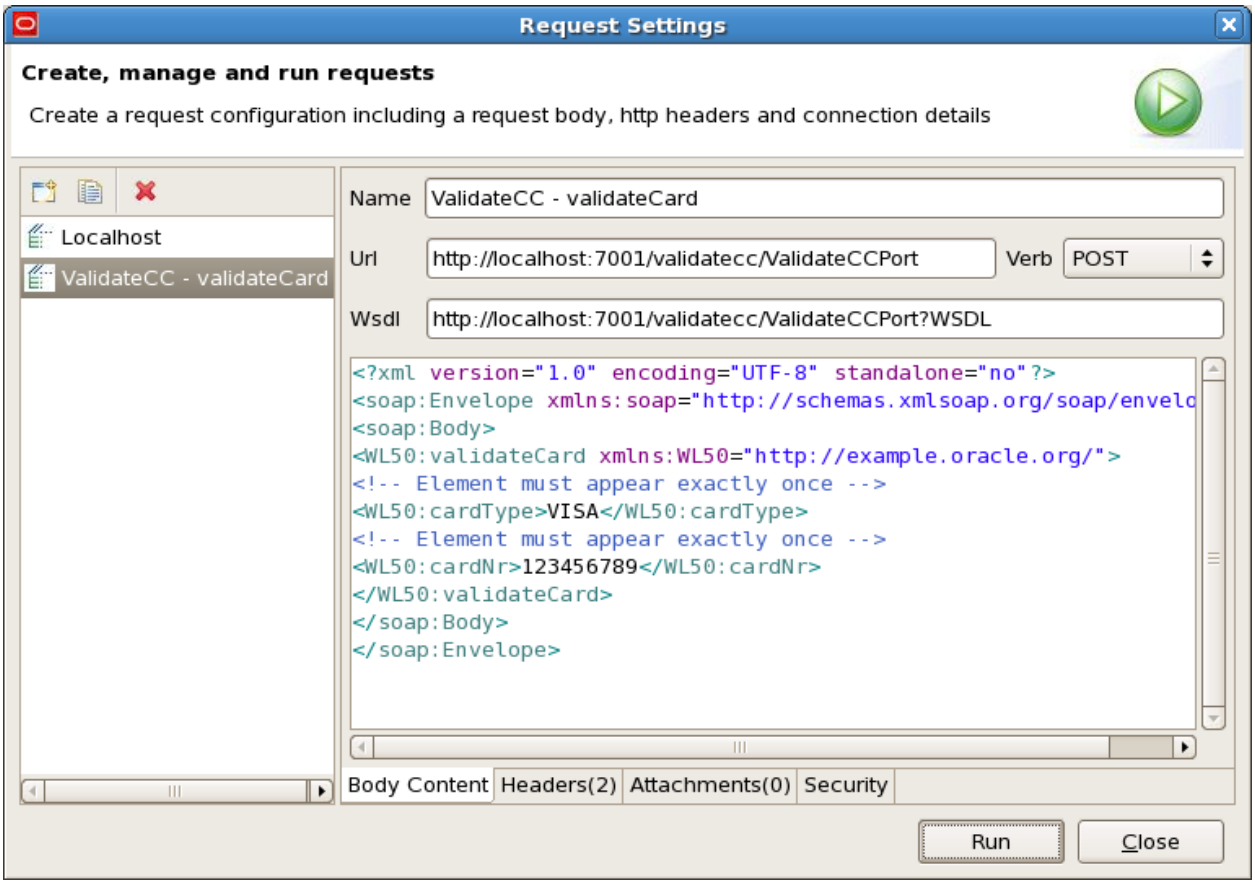
3. First, you send the request from Service Explorer directly to the back-end service by performing the following steps:

a. Click the inverted triangle in the toolbar as depicted below, and select “Request Settings…” from the drop-down menu.



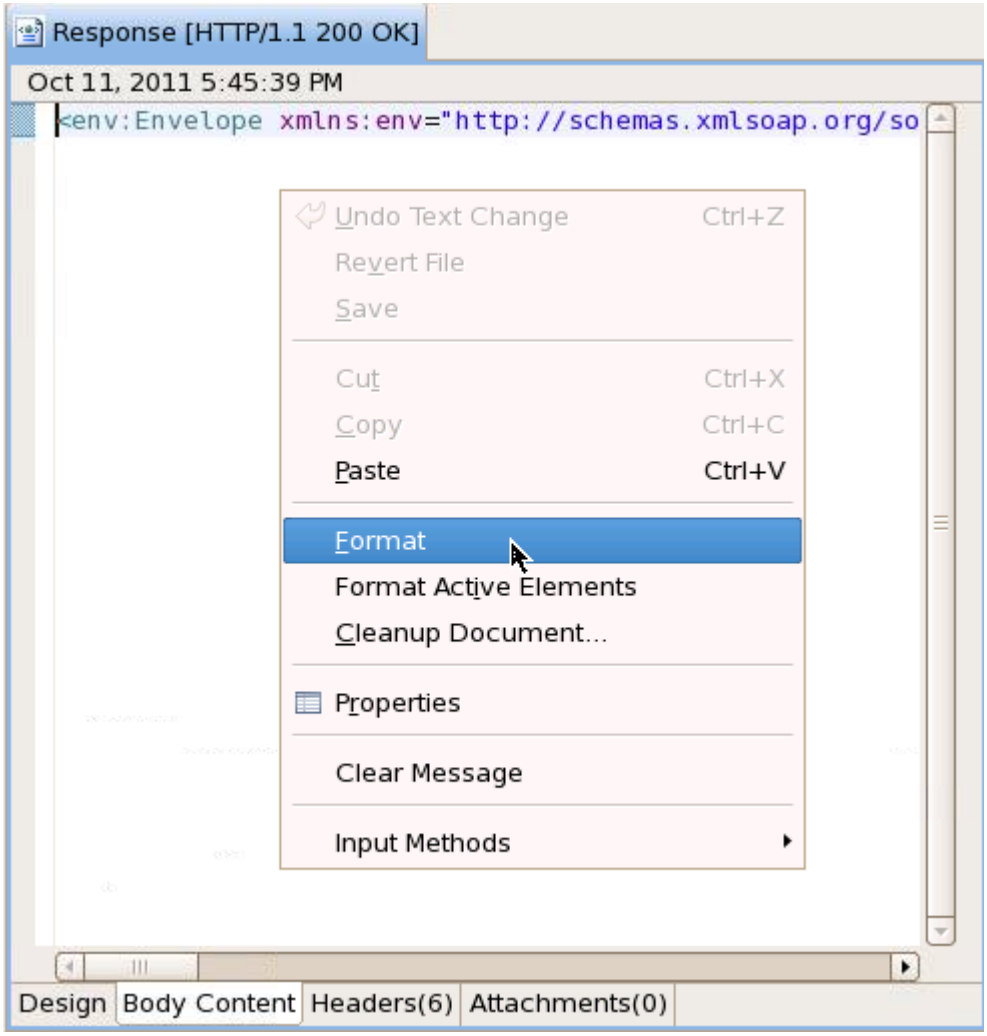
b. In the Request Settings window, make sure the URL that serves the request is: `http://localhost:7001/validatecc/ValidateCCPort`

c. Provide a valid `cardType` and card number in the request message body. Use the following screenshot as a guide:

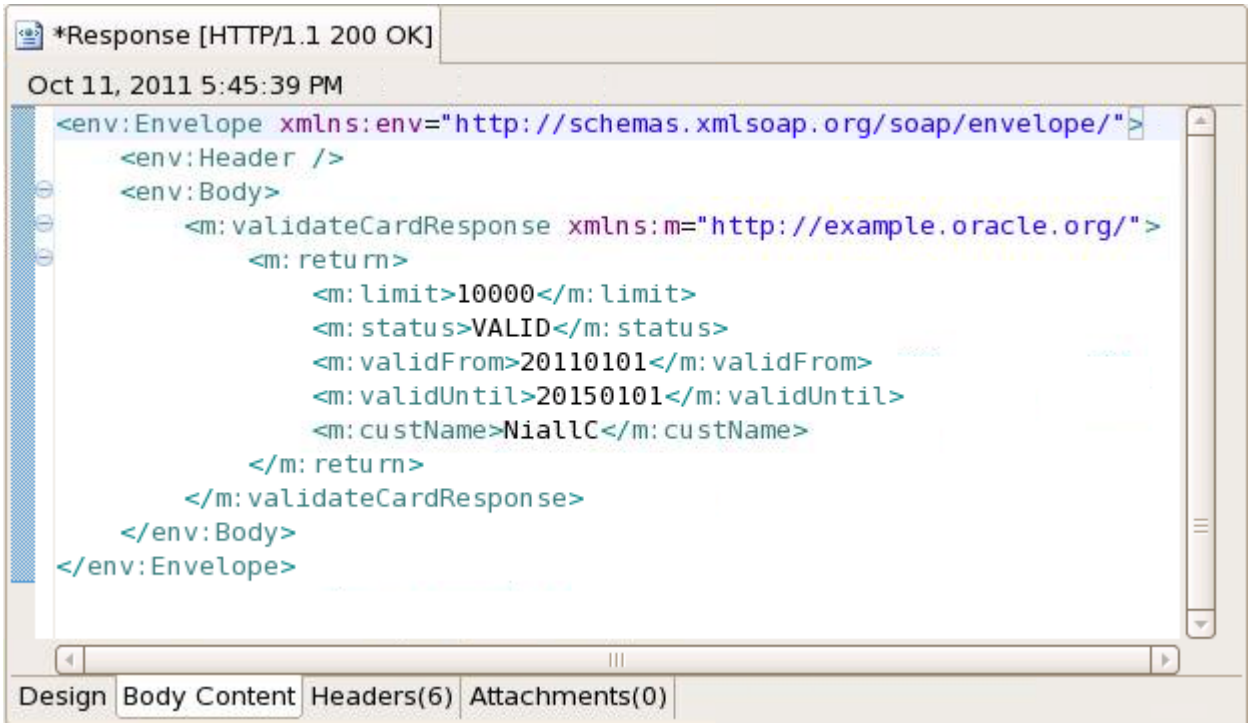


d. Click Run. You should see a response message displayed in the Response tab on the right.

e. To format the message, right-click the Response pane, and select Format.

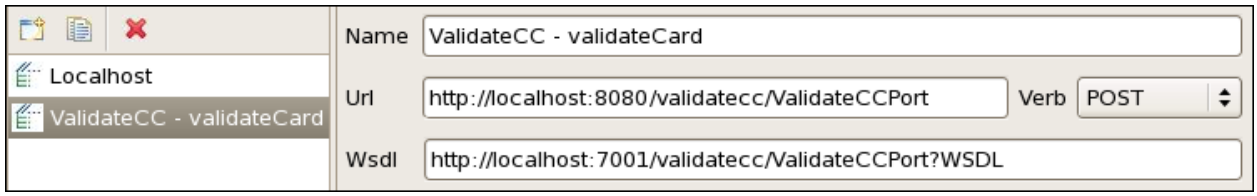


The response message panel is updated with the reformatted content as shown in the screenshot below:



4. Next, you send the request from Service Explorer to the Gateway, which then passes down to the back-end service:
   a. Similar to the previous step, click the inverted triangle in the toolbar as depicted below, and select “Request Settings…” from the drop-down menu.
   b. In the Request Settings window, change the URL (port number) that serves the request to:

      `http://localhost:8080/validatecc/ValidateCCPort`



   c. Provide a valid `cardType` and card number in the request message body, or you can use the existing values.
   d. Click Run. You should see a response message displayed in the Response tab on the right.
5. When you are done, leave the Service Explorer window open. You will need it in the next practice.

Practices for Lesson 5: Monitoring, Logging, and Tracing

Chapter 5

Practices for Lesson 5: Overview

Practices Overview

In these practices, you learn how to monitor the traffic of service requests that go through Enterprise Gateway by using various OEG tools.

Practice 5-1: Enable Monitoring

Overview

In this practice, you make sure that monitoring is enabled for Traffic Monitor, the Real time monitoring console, and Service Monitor in Policy Studio.

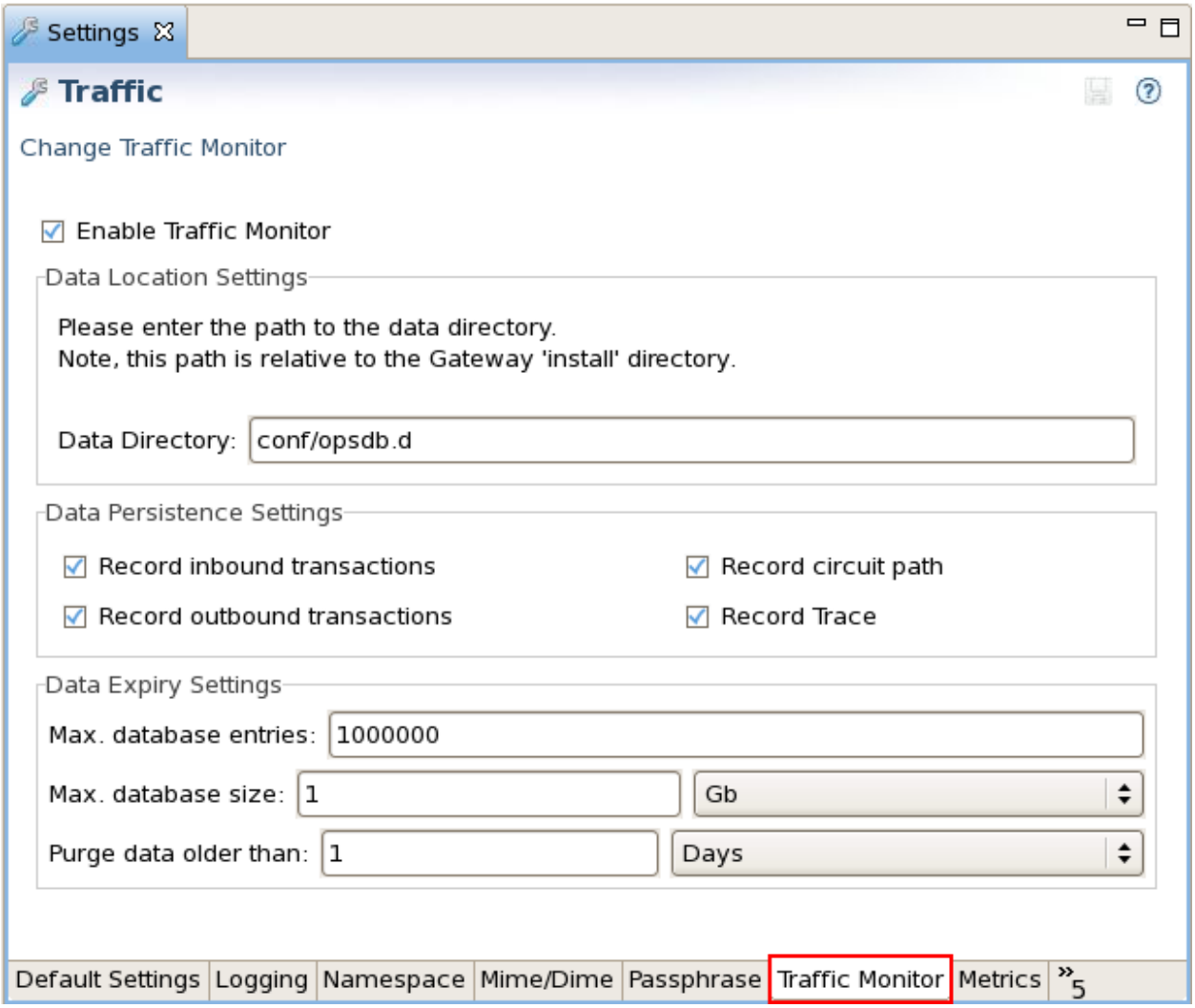
Assumptions

The Enterprise Gateway and WebLogic Admin server are up and running.

Tasks

1. Open Policy Studio from the desktop if it is not open.
2. To ensure that Traffic Monitor has been enabled for the Enterprise Gateway process in Policy Studio, perform the following steps:
   a. In the left navigation pane, select Settings. The Default Settings screen appears in the right pane.
   b. Click the Traffic Monitoring tab at the bottom of the page. The settings allow you to configure the web-based Traffic Monitor tool and its message traffic log.

c. Make sure “Enable Traffic Monitor” is selected. This enables the web-based Traffic Monitor tool.



3. To ensure that monitoring is enabled for the Real time monitoring console, in addition to traffic monitoring being enabled (as described in the previous step), you also need to enable real-time monitoring by performing the following steps:
   a. Click the Metrics tab at the bottom of the page.
   b. Make sure “Enable real time monitoring” is selected. This setting enables real-time monitoring globally for Enterprise Gateway.

Practice 5-2: Monitoring using Traffic Monitor and Real Time Monitoring Console

Overview

In this practice, you will monitor the traffic going through the Gateway by using the Traffic Monitor and Real time monitoring console, available through the main product Welcome page hosted at http://localhost:8090.

Assumptions

The Enterprise Gateway and WebLogic Admin server are up and running.

Tasks

Your tasks here are to:

- Send a few test messages to the back-end web service
- Examine the Message Traffic Log in Traffic Monitor
- View the monitoring statistics in the Real time monitoring console

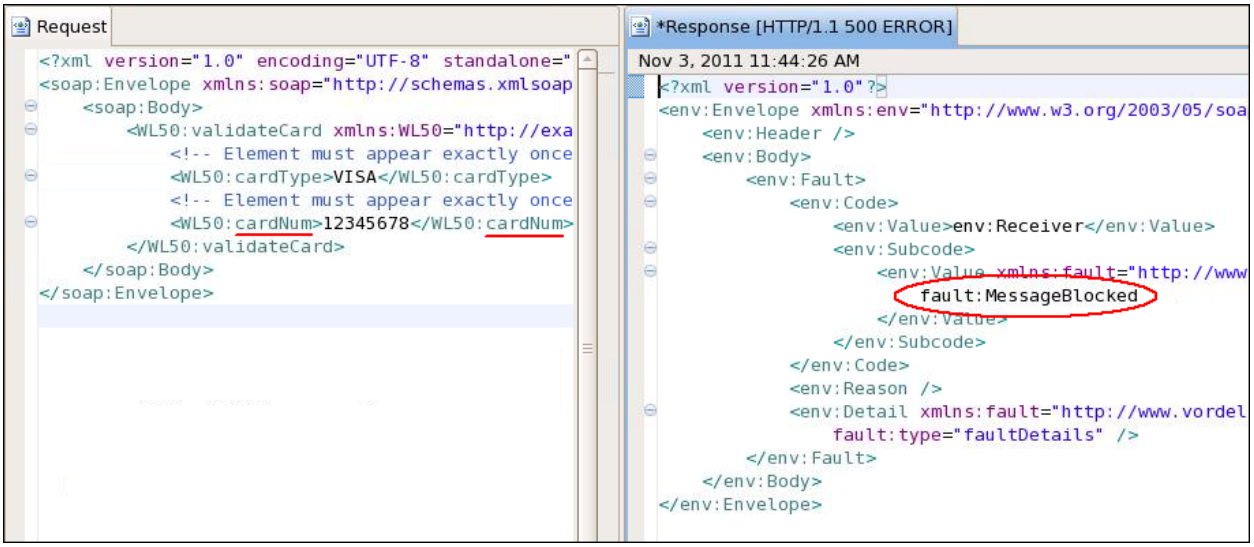
1. Open Service Explorer; select ValidateCC – validateCard request configuration.
2. Click the Run button (green “Play” button on the toolbar) to send the test message to the back-end web service through Enterprise Gateway.

3. To create some traffic, you can change the request message by modifying the credit card type and number, and then click Run to send the requests to the target web service.

   Note: The valid credit card types are VISA and AMEX.

Testing Schema Validation

1. Change the request message to include the attribute <cardNum> instead of <cardNr> so that it remains valid XML (opening tag matches closing tag) but fails schema validation. The change is shown below:



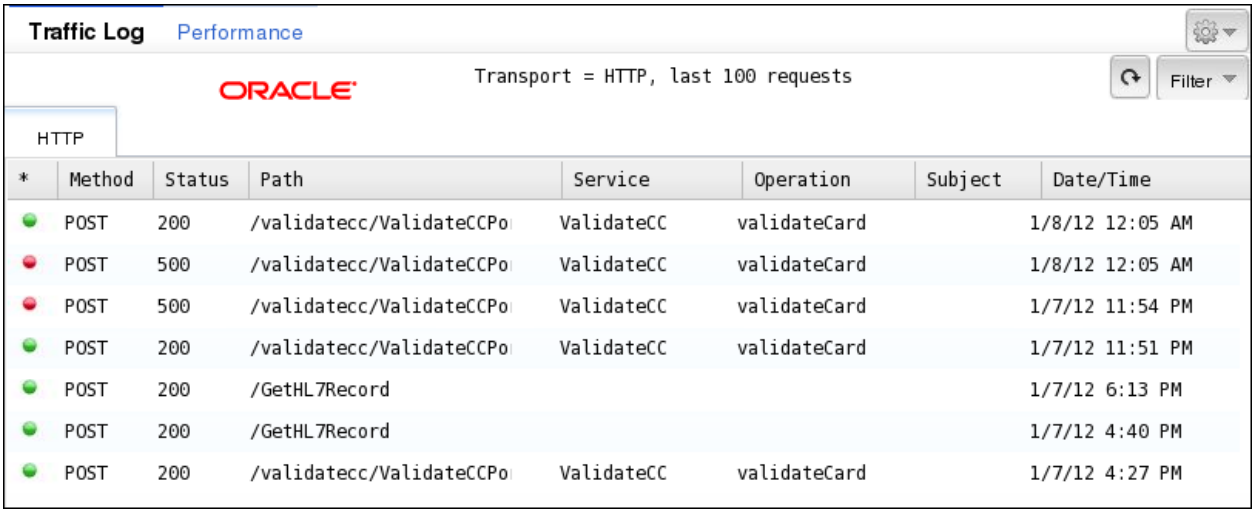
2. Send the request again by clicking the Run button. You should get a 500 error response. This message is blocked, because the schema validation fails. You will see this in the Traffic Monitor and Real time monitoring console later in this practice.

Monitoring with Traffic Monitor

1. Open the Firefox browser by using the desktop icon, and enter the following URL in the address field:

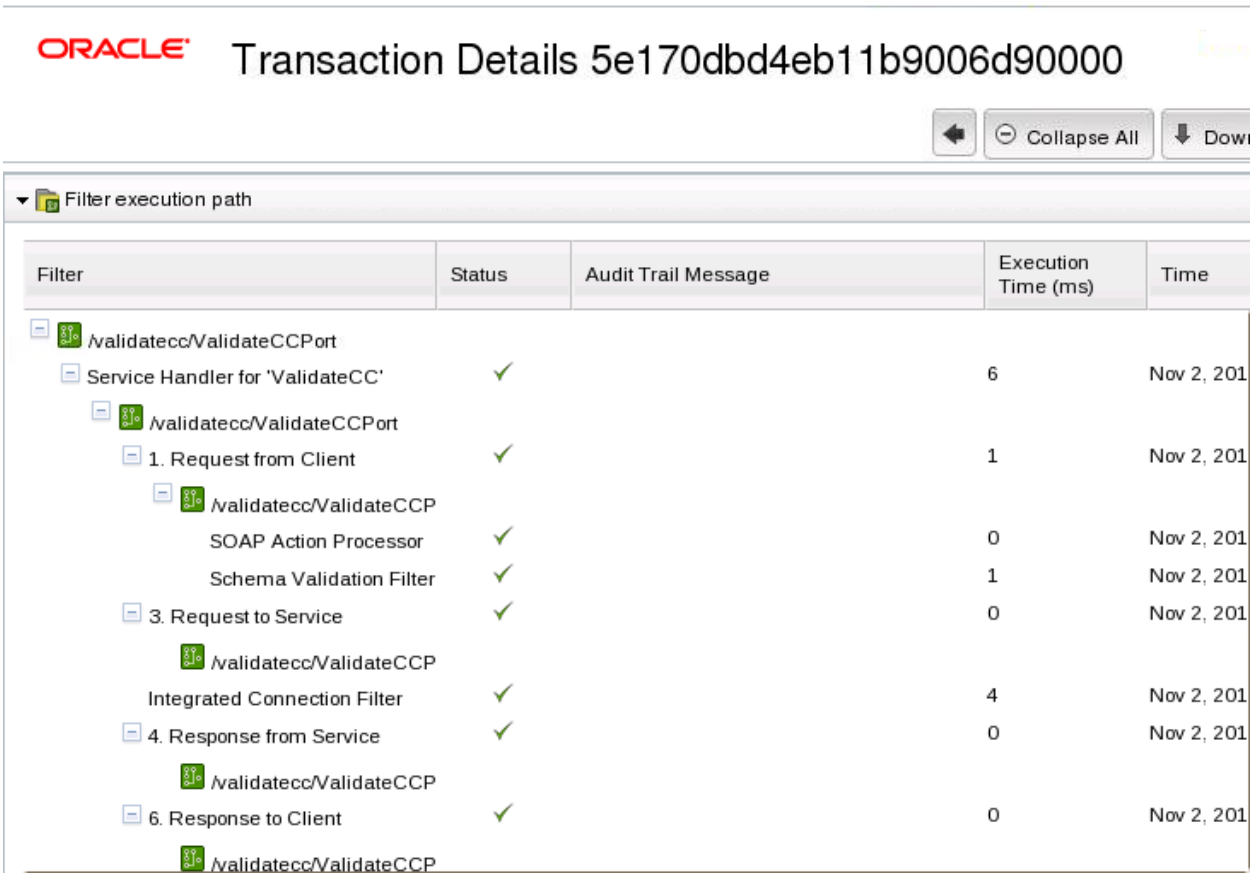
   `http://localhost:8090`

2. When prompted, log in with username `admin` and password `changeme`.
3. In the Welcome to Oracle Enterprise Gateway home page, click the “Traffic Monitor” link. You should see a list of messages that you sent earlier by using Service Explorer.



*	Method	Status	Path	Service	Operation	Subject	Date/Time
●	POST	200	/validatecc/ValidateCCPo	ValidateCC	validateCard		1/8/12 12:05 AM
●	POST	500	/validatecc/ValidateCCPo	ValidateCC	validateCard		1/8/12 12:05 AM
●	POST	500	/validatecc/ValidateCCPo	ValidateCC	validateCard		1/7/12 11:54 PM
●	POST	200	/validatecc/ValidateCCPo	ValidateCC	validateCard		1/7/12 11:51 PM
●	POST	200	/GetHL7Record				1/7/12 6:13 PM
●	POST	200	/GetHL7Record				1/7/12 4:40 PM
●	POST	200	/validatecc/ValidateCCPo	ValidateCC	validateCard		1/7/12 4:27 PM

4. In the Traffic Log page, click a message entry to view the transaction details. It resembles the following screenshot:

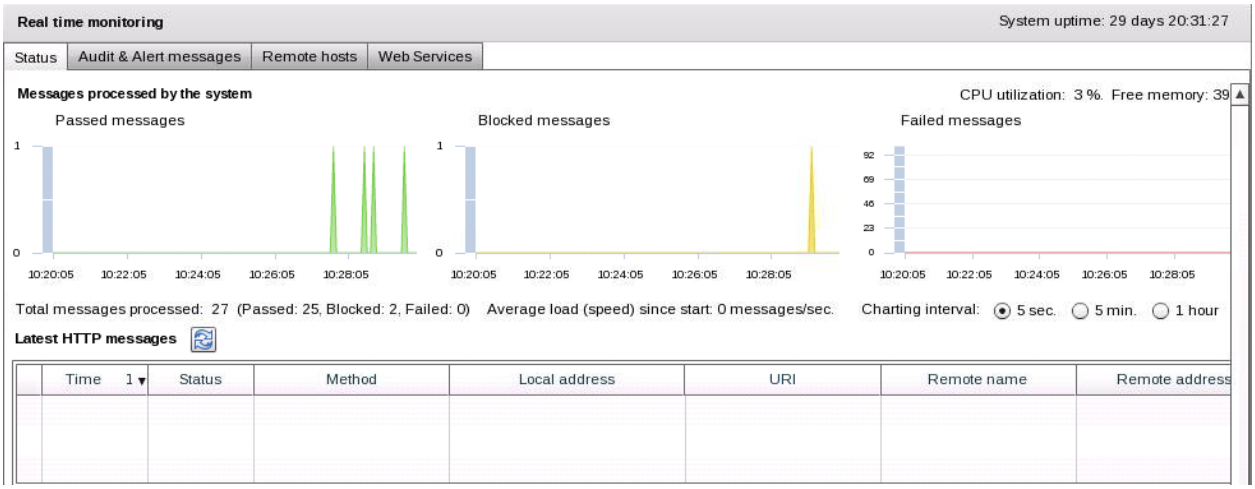


You should see:

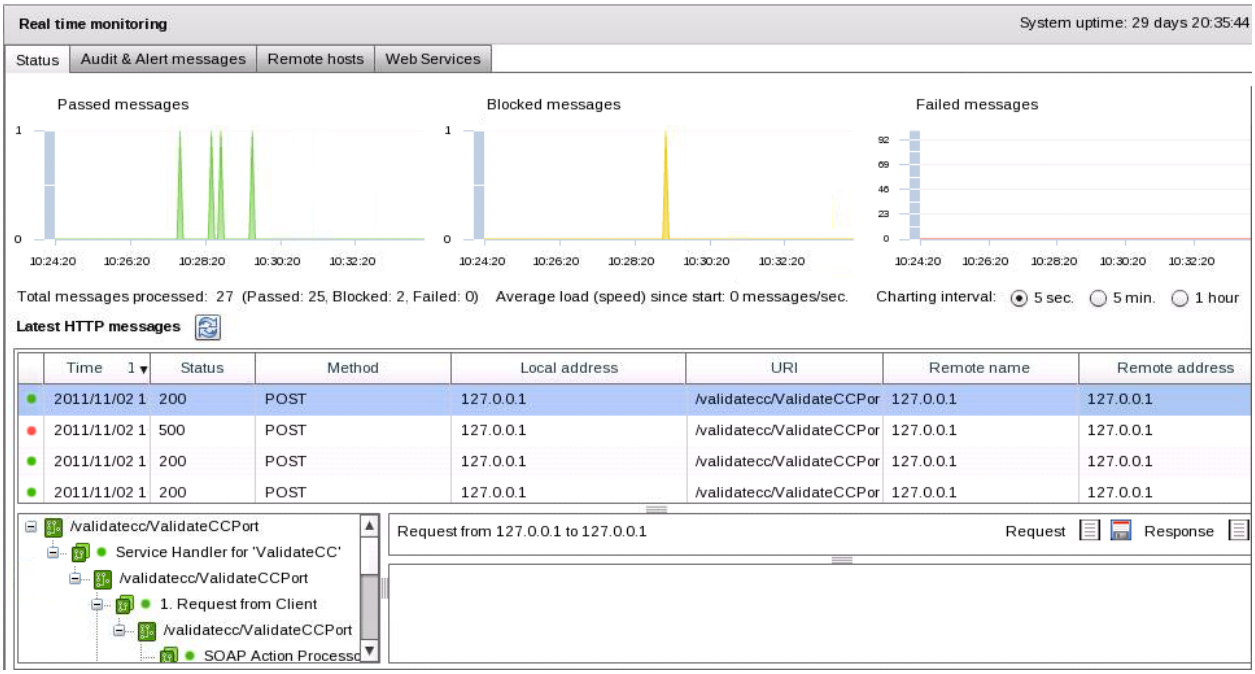
- The filters that the message passes through in the policy circuit on the top of the screen
- The contents of the request message from the client and the response message from Enterprise Gateway at the bottom of the screen
- The content of the request message from Enterprise Gateway and the response message from the web service at the bottom of the screen

Monitoring with Real Time Monitoring Console

1. Use the Back button of the browser to return to the Welcome to Oracle Enterprise Gateway home page. Click the “Real time monitoring console” link. You are presented with a graph showing passed, blocked, and failed messages. Use the following screenshot as a guide:

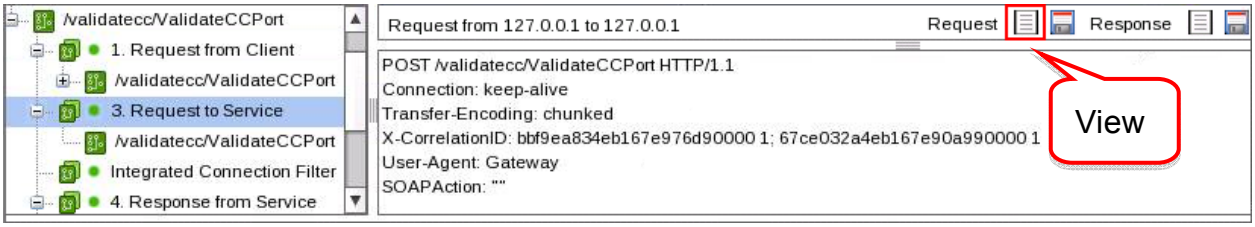


2. Click the Refresh button  next to “Latest HTTP message” to show a list of transactions. You can select an individual message to view the message path and the contents of each request message and response message.



3. Select the message whose contents you wish to view. This displays the message path for the selected message in the bottom-left pane.

4. Select the inbound message from the Enterprise Gateway, which is “3. Request to Service” in the message path. And click the View button for the Request message to view its contents.



Questions:

- Is the list of transactions displayed in the Real time monitoring console the same as what you see in the Message Traffic Log page?
- Do you see any differences of the transaction details presented between the console and Traffic Monitor?

Practice 5-3: Viewing Reports in Service Monitor

Overview

In this practice, you start the Service Monitor component, configure Service Monitor settings to enable real-time monitoring data to be written to the database, and then view the metrics data in the browser.

Assumptions

The Enterprise Gateway and WebLogic Admin server are up and running.

Tasks

Starting Service Monitor

1. Locate the “Start Service Monitor” launcher on the Desktop, and double-click it.
2. In the terminal window, wait until you see a message similar to:

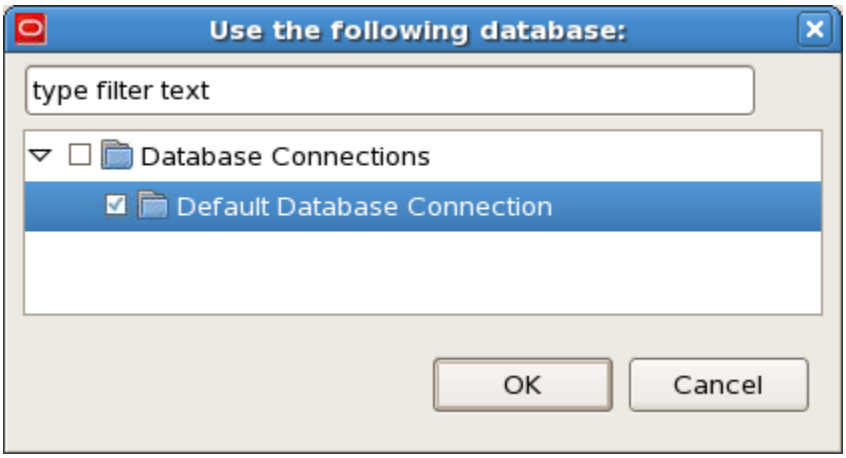
```
...
INFO  06/Jan/2012:10:30:39.466 [260538f0]  TCP interface
INFO  06/Jan/2012:10:30:39.466 [260538f0]  checking invariants
for interface *:8040
INFO  06/Jan/2012:10:30:39.466 [260538f0]  listen on address
0.0.0.0/8040
INFO  06/Jan/2012:10:30:38.573 [260538f0]  Initializing Server
Servlet
INFO  06/Jan/2012:10:30:38.573 [260538f0]  Starting
ESSOAPProvider with entitystore
...
INFO  06/Jan/2012:10:30:38.573 [260538f0]  ... monitoring
started
INFO  06/Jan/2012:10:30:38.838 [260538f0]  Initializing
Configuration Servlet
INFO  06/Jan/2012:10:30:39.627 [260538f0]  Starting
ESSOAPProvider with entitystore
...
INFO  06/Jan/2012:10:30:39.628 [260538f0]  service started
(version 6.3.0-2011-12-20, pid 6055)
INFO  06/Jan/2012:10:30:39.633 [260538f0]  loaded netservice
library
```

   Note that Service Monitor runs on port 8040.
3. Minimize the oegservicemonitor terminal window.

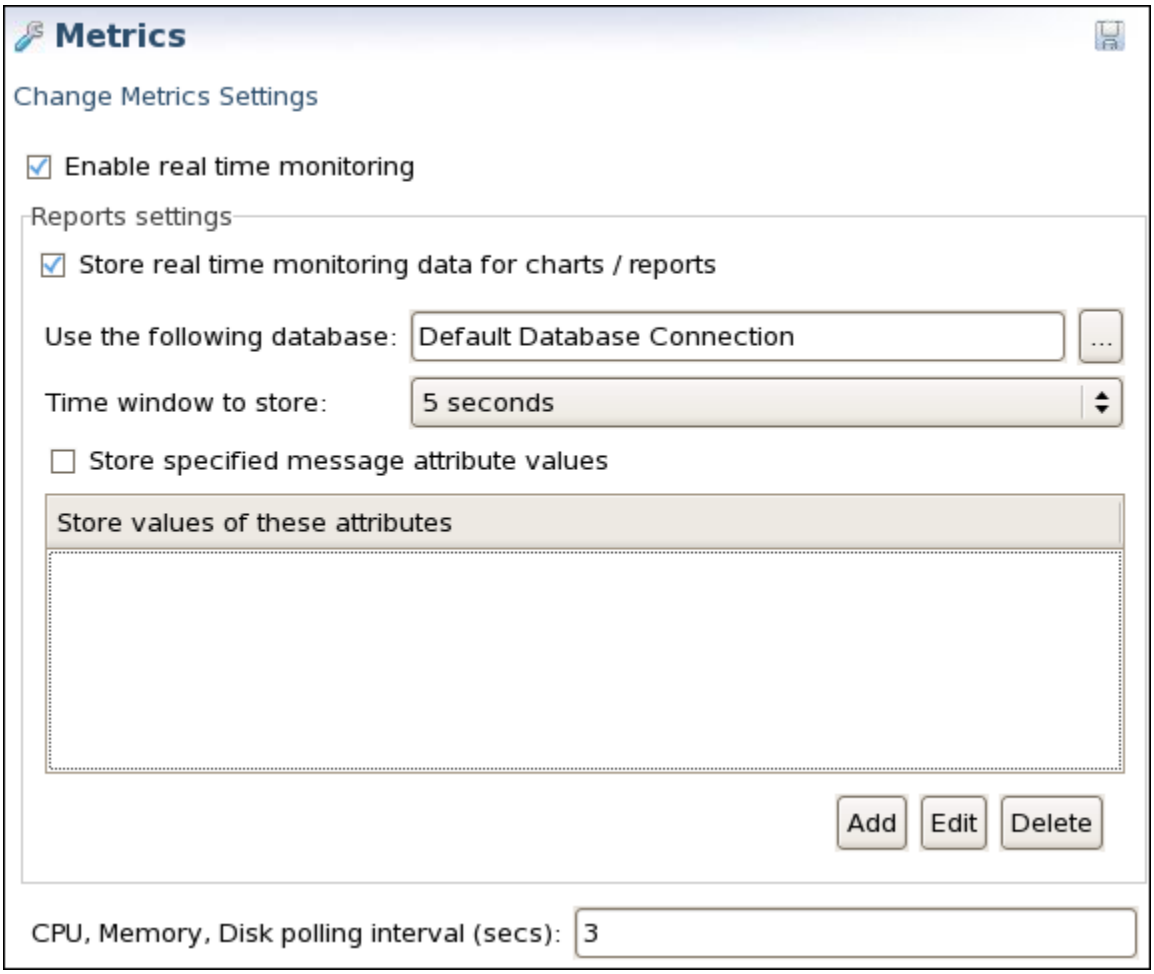
Configuring Service Monitor Settings

1. Open Policy Studio. In the left navigation pane, select Settings.
2. Click the Metrics tab at the bottom of the page.
3. Select the “Store real-time monitoring data for charts / reports” option.

4. Use Default Database Connection as the database for metrics data.



5. The Metrics setting is updated with your selected database:



6. Save your settings by clicking the Save button on the top-right of the screen.
7. Deploy the configuration by pressing F6.

Viewing Reports in Service Monitor

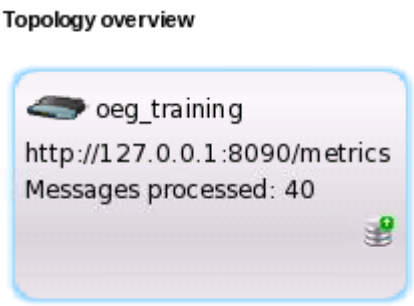
1. Open the Firefox browser by using the desktop icon, and enter the following URL in the address field:

```
http://localhost:8040
```

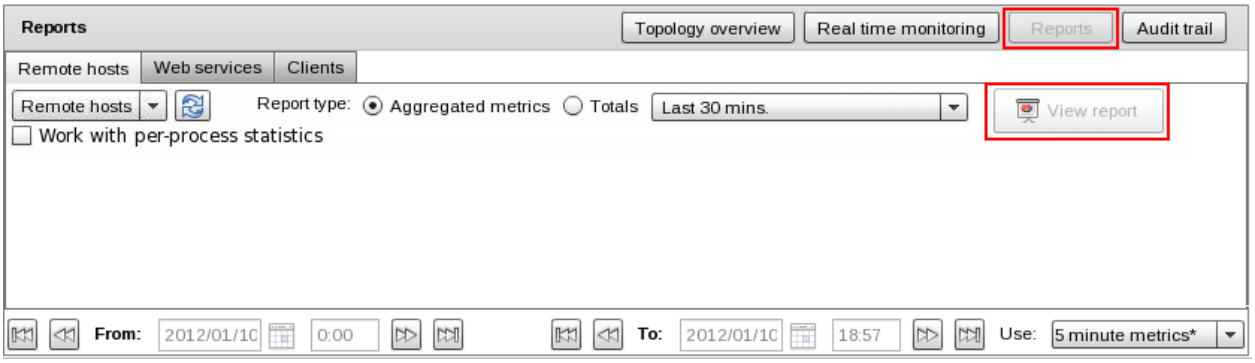
2. When prompted, log in with username `admin` and password `changeme`. You are presented with the Topology overview page without a gateway configured yet.
3. To add the Enterprise Gateway instance as a node, perform the following steps:
   a. Click the Add Gateway button at the bottom of the page.
   b. Configure the following fields in the Node Settings dialog box:

Field	Choices or Values
Server URL	Accept the default: http://127.0.0.1:8090/metrics
Username	`admin`
Password	`changeme`
Node name	`oeg_training`

   c. Click OK. You should see the oeg_training node presented in the Topology overview page as below:

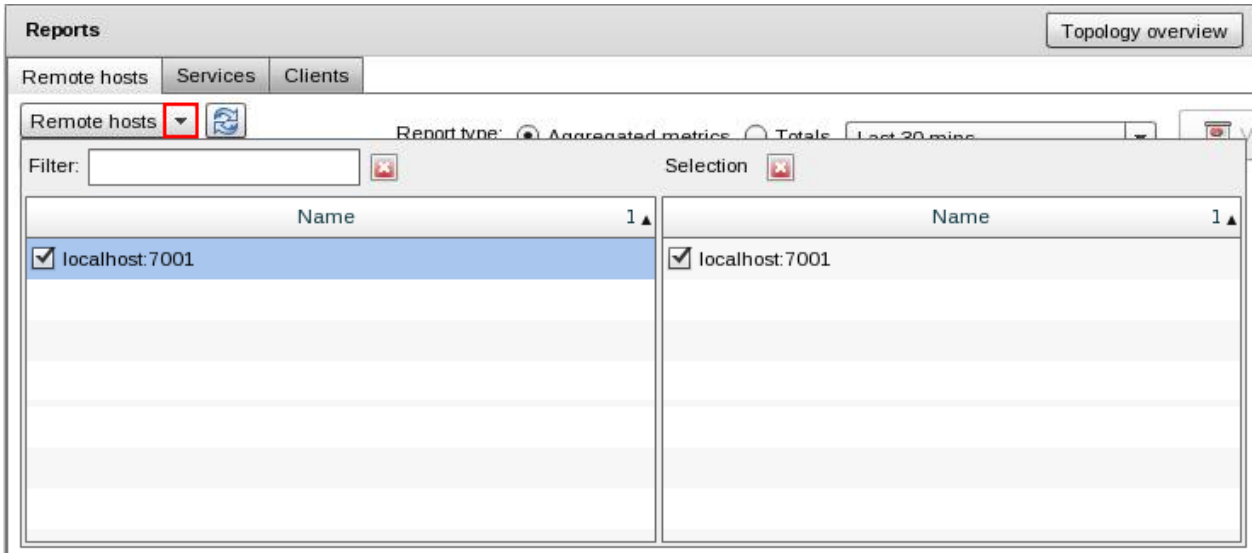


4. Click the Reports button in the toolbar on the top-right of the page.

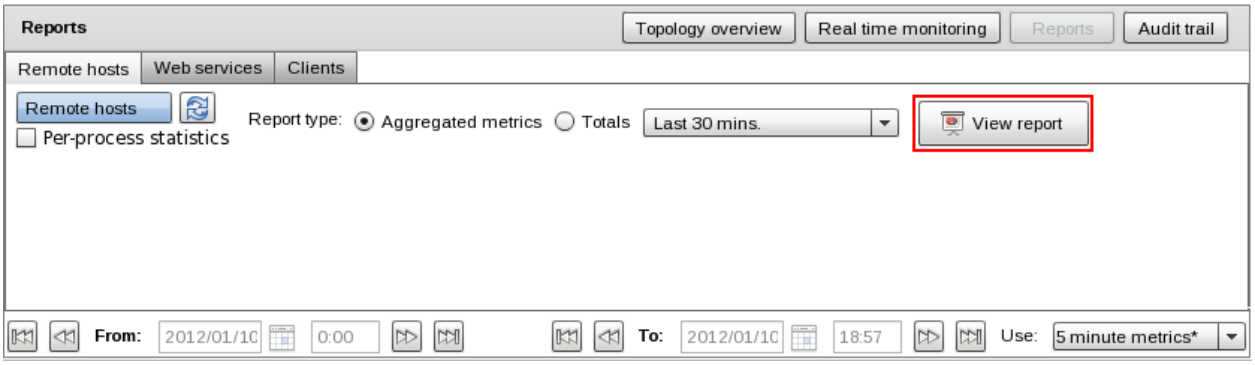


Note that the View Report button is disabled.

5. To enable View Report, you need to select the remote hosts to get the metrics data. Click the Remote hosts drop-down menu and select localhost:7001 from the list.

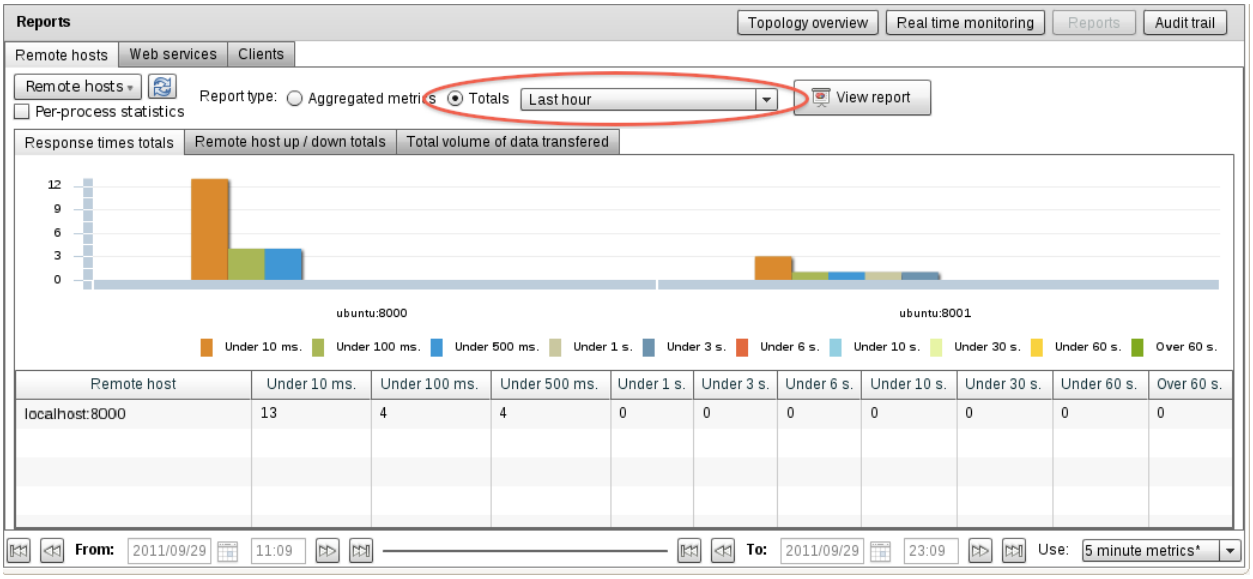


Now the View Report button becomes enabled.



6. Make sure the graph interval at the bottom right is set to 5-minute metrics.
7. Select the Totals option for Report type, and select “Last Hour” in the drop-down.

8. Click View Report. A report similar to the one below is generated:



9. When you are done, you can stop the Service Monitor process.

Practice 5-4: Configuring Logging and Trace

Overview

In this practice, you configure log output location, and examine the log and trace file.

Assumptions

The Enterprise Gateway and WebLogic Admin server are up and running.

Tasks

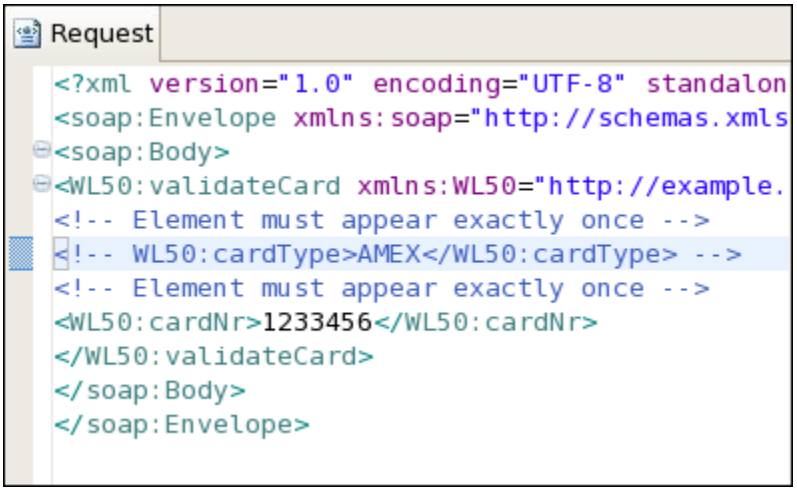
Configuring Log output

1. Open Policy Studio and select Settings in the left frame. The Default Settings page is displayed in the right frame.
2. Select the Logging tab at the bottom. The Logging home page appears, showing the various options to enable and configure logging.
3. In the Text File tab, select the check box “Enable logging to file,”, and accept the default settings. Use the contextual help if you need details.
4. Click the Save Settings button on the top right of the page.

5. Deploy the configuration (F6).

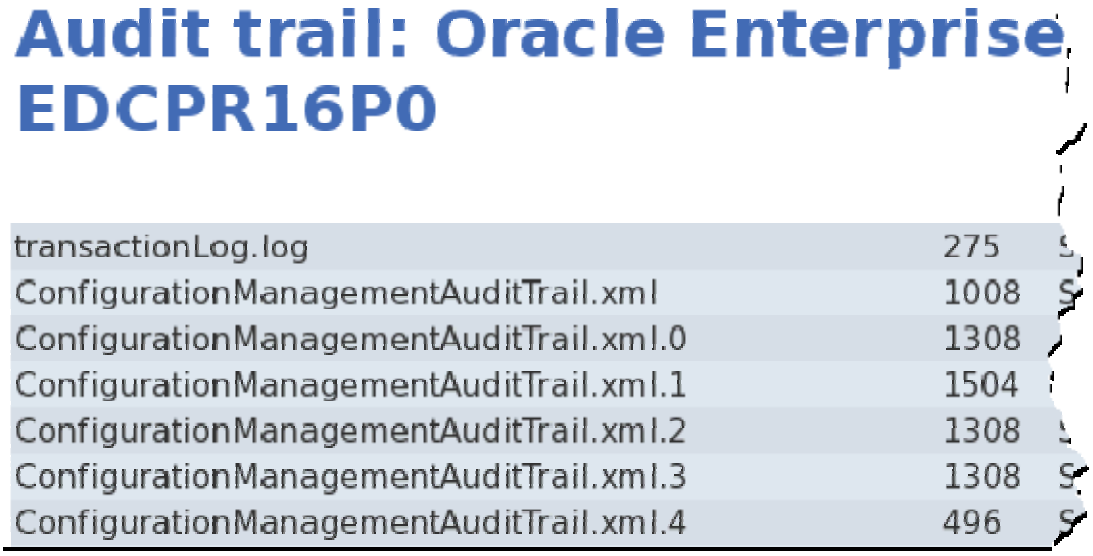
Testing the service

1. Open Service Explorer and select ValidateCC – validateCard request configuration.
2. Comment out the line with the `cardType` element. Use the following image as a guide:



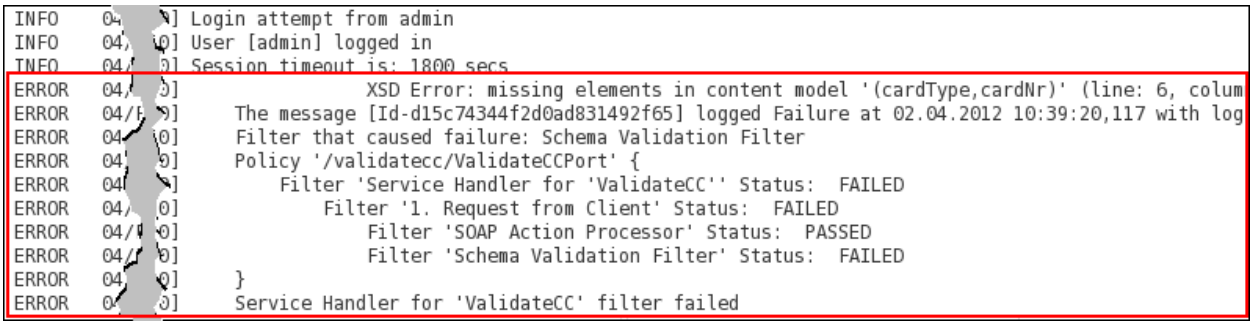
3. Click Run. You should see a response message displayed with a 500 error.
4. Open the Firefox browser, log in to http://localhost:8090/

5. On the Enterprise Gateway home page, click the “Audit trail log files” link. You should see the list of audit trails and newly added log file.



6. Click the `transactionLog.log` file; you should see a log entry as below:

7. To get detailed information, you need to see the trace file. To do so:
   a. Click the Back button of the browser to go to the Enterprise Gateway home page.
   b. Click the Trace Files link.
   c. On the Trace Files home page, click `OracleEnterpriseGateway.trc`, the latest trace file.
   d. Use the time stamp of the error log entry in the `transactionLog` file, for example 00:10:56, to locate the error in the trace file. You should see the details of the error.



8. Go back to Service Explorer, remove the comment in the request, and send the request again. You should see the response without any error.
9. Go to the browser, and open the `transactionLog` file. Do you see a new log entry? If not, can you explain why?

Practices for Lesson 6: Managing Configurations

Chapter 6

Practices for Lesson 6: Overview

Practices Overview

In these practices, you will version the Gateway instance (process) configuration, and export some configuration data to a local XML file.

Practice 6-1: Versioning Process Configuration

Overview

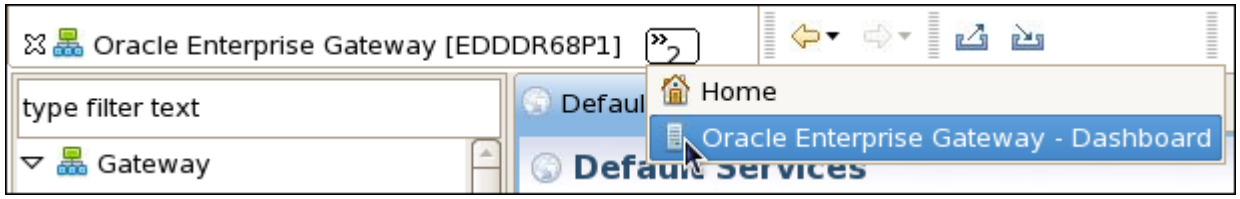
In this practice, you will add a version to your gateway configuration.

Assumptions

Enterprise Gateway is up and running.

Tasks

1. To open Oracle Enterprise Gateway Dashboard:
   - If your Policy Studio is open, and you are on the Active Server Configuration screen, click Show List, and select Oracle Enterprise Gateway – Dashboard.

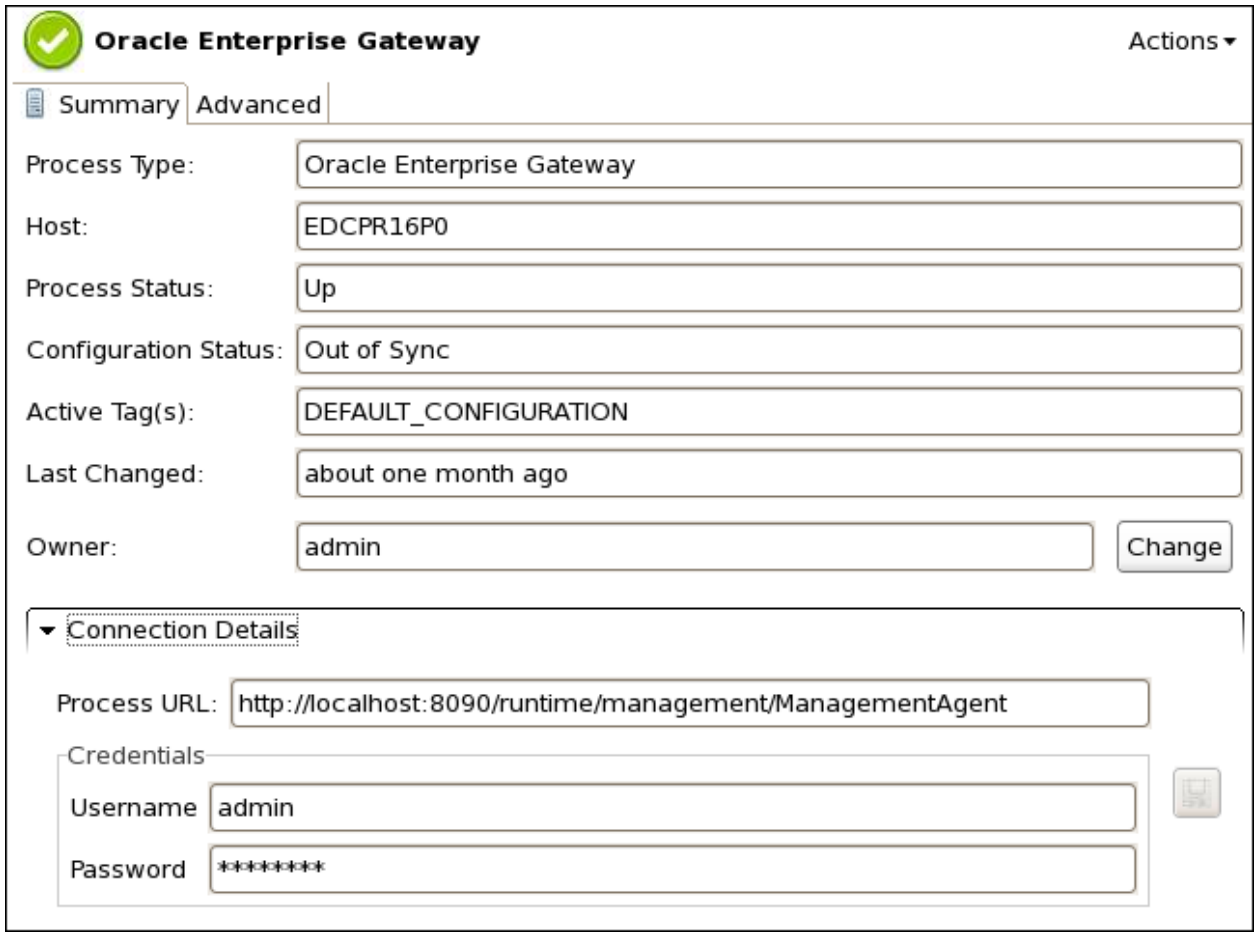


   - If your Policy Studio is closed, start it by using the shortcut on the desktop.

   a. On the Home tab, connect to OEG Gateway by clicking “Enterprise Gateway – localhost”.

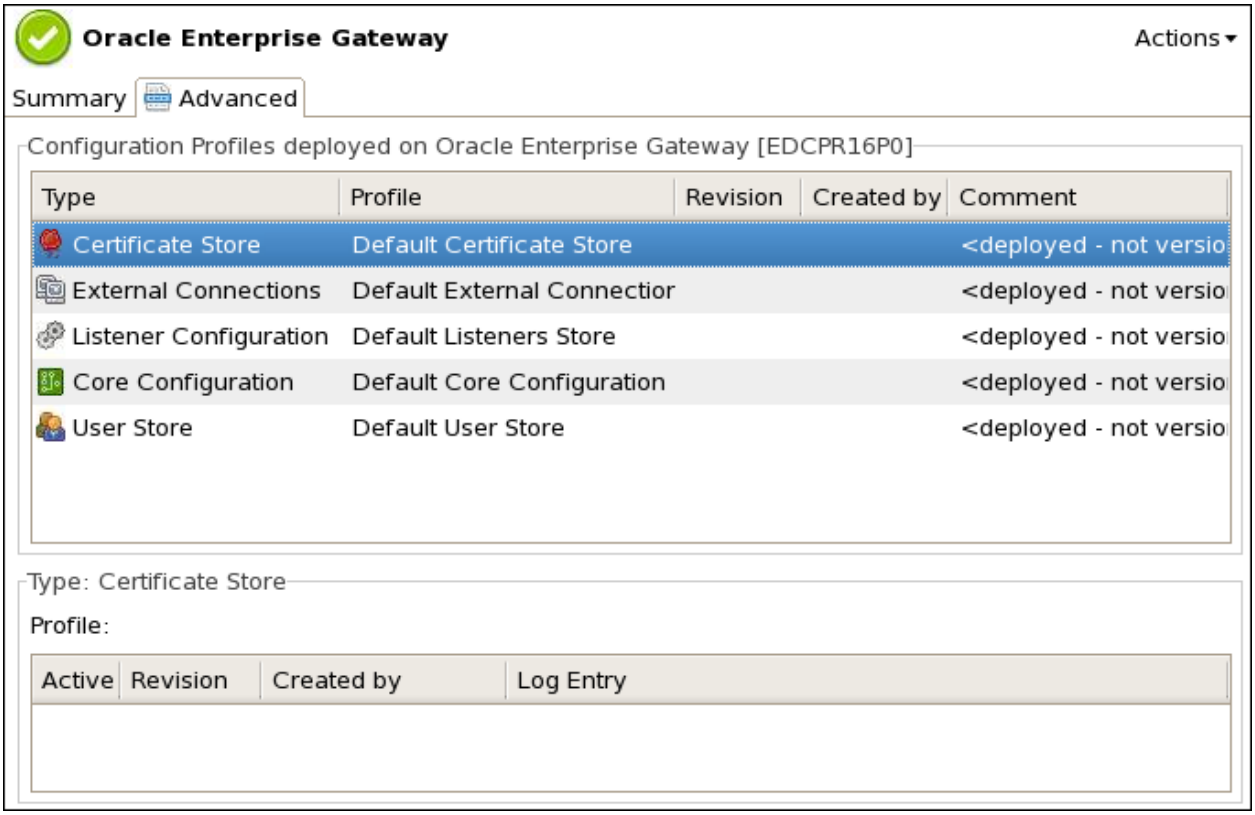
   b. In the Open Connection dialog box, accept the default settings and click OK.

2. In the Oracle Enterprise Gateway Dashboard, you can view the process details that are displayed in the right frame:
   a. On the Summary tab, click the Connection Details link at the bottom to view details such as the URL, username, and password for the process.



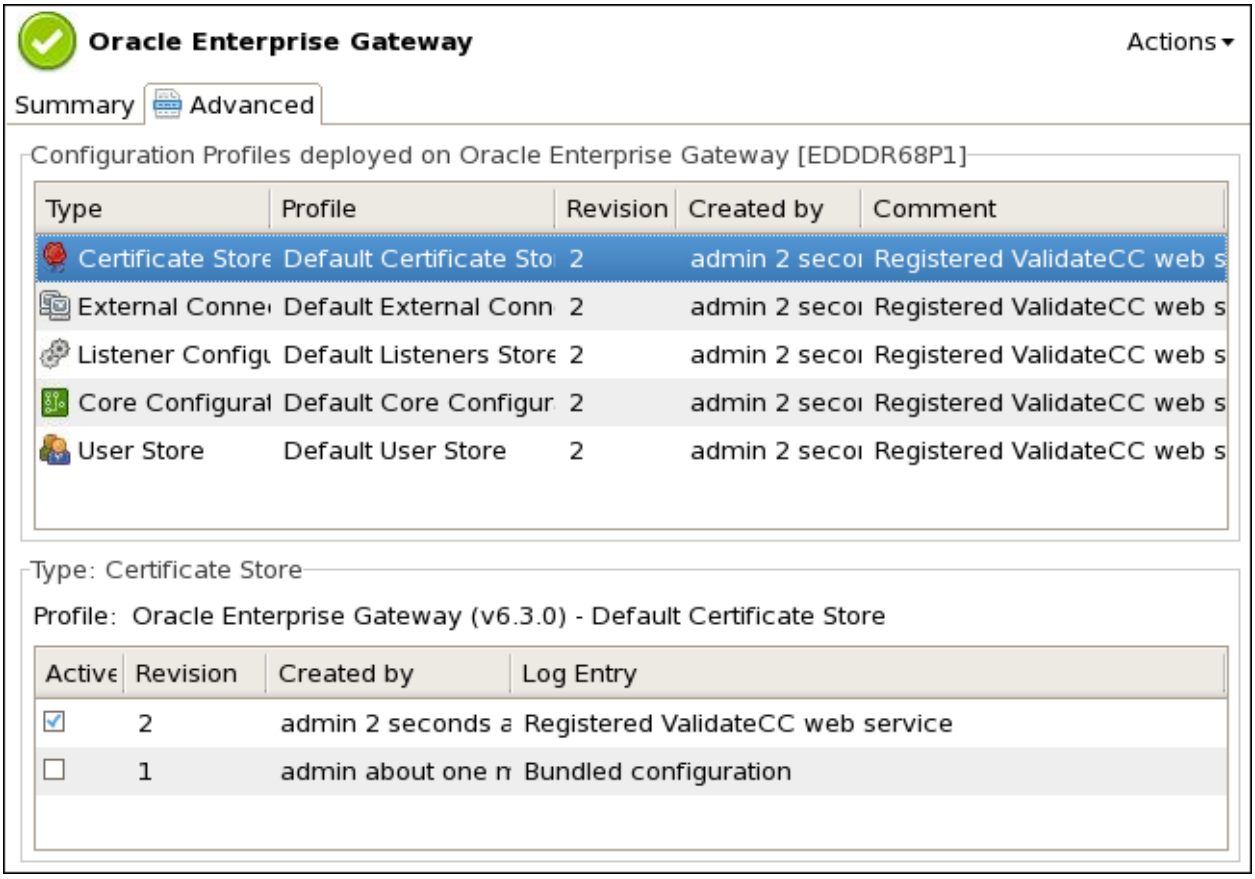
Note that the Configuration Status is Out of Sync. If you have edited an active process configuration, and deployed updates to the server without versioning, the Configuration Status for the selected process is displayed as Out of Sync in the Summary tab.

b. Click the Advanced tab. You should see component configuration stores that are deployed on the process, as shown below:



3. To version the configuration of the process, Oracle Enterprise Gateway [*hostname*], performs the following steps:
   a. Click Actions, and select Version Configuration from the drop-down menu. Alternatively, right-click the process in the Process List, and select Version Configuration.

b. Enter a Comment like “Registered ValidateCC web service” in the Versioning dialog box, and click Yes. The Advanced tab is updated with the versioning information as shown below:



4. Click the Summary tab, and note that the Configuration Status for the selected process is now displayed as Deployed. The Version Configuration link is no longer displayed in the Action menu.

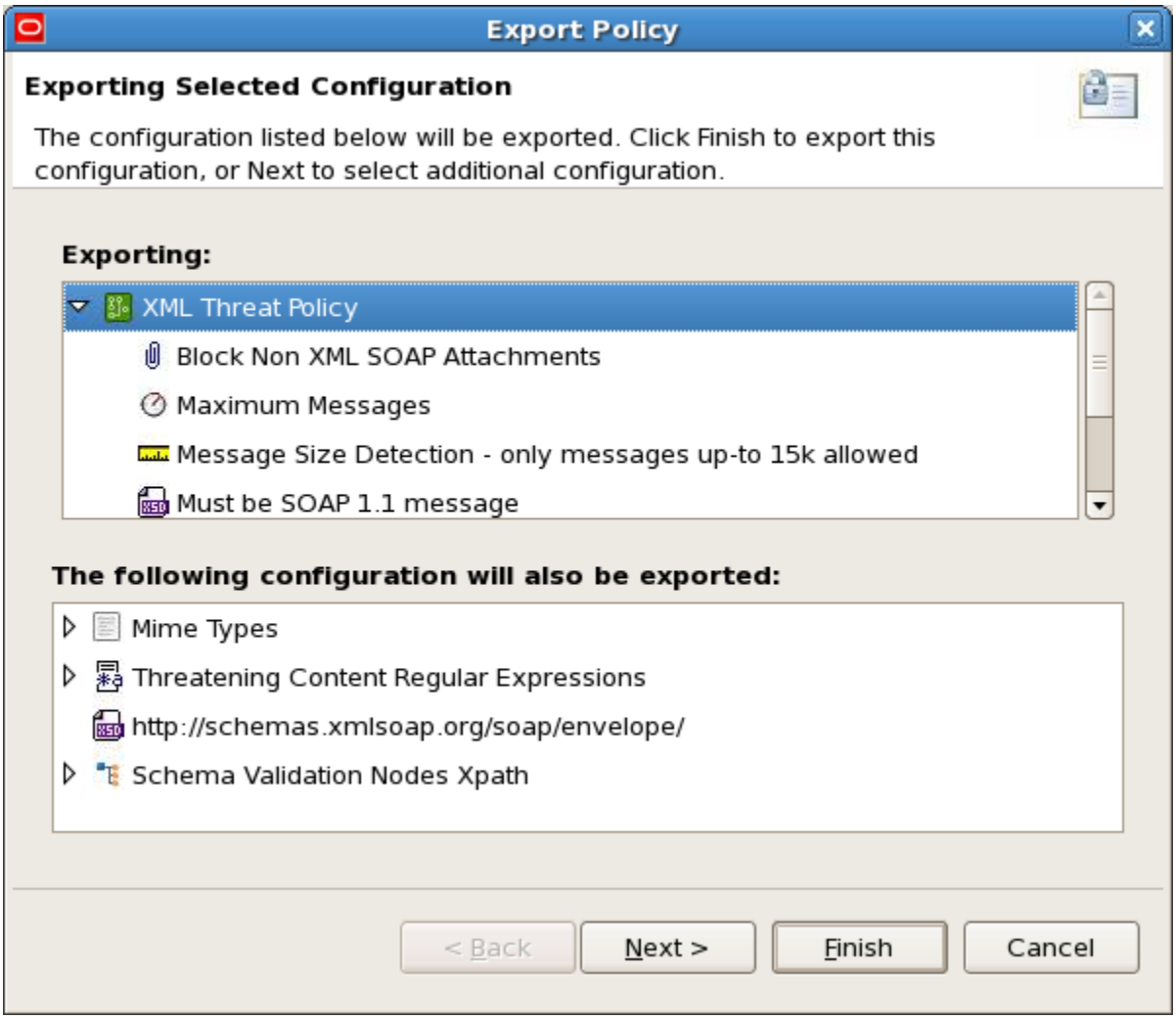
Practice 6-2: Exporting the Configuration Data

Overview

In this practice you export a sample policy out of your configuration.

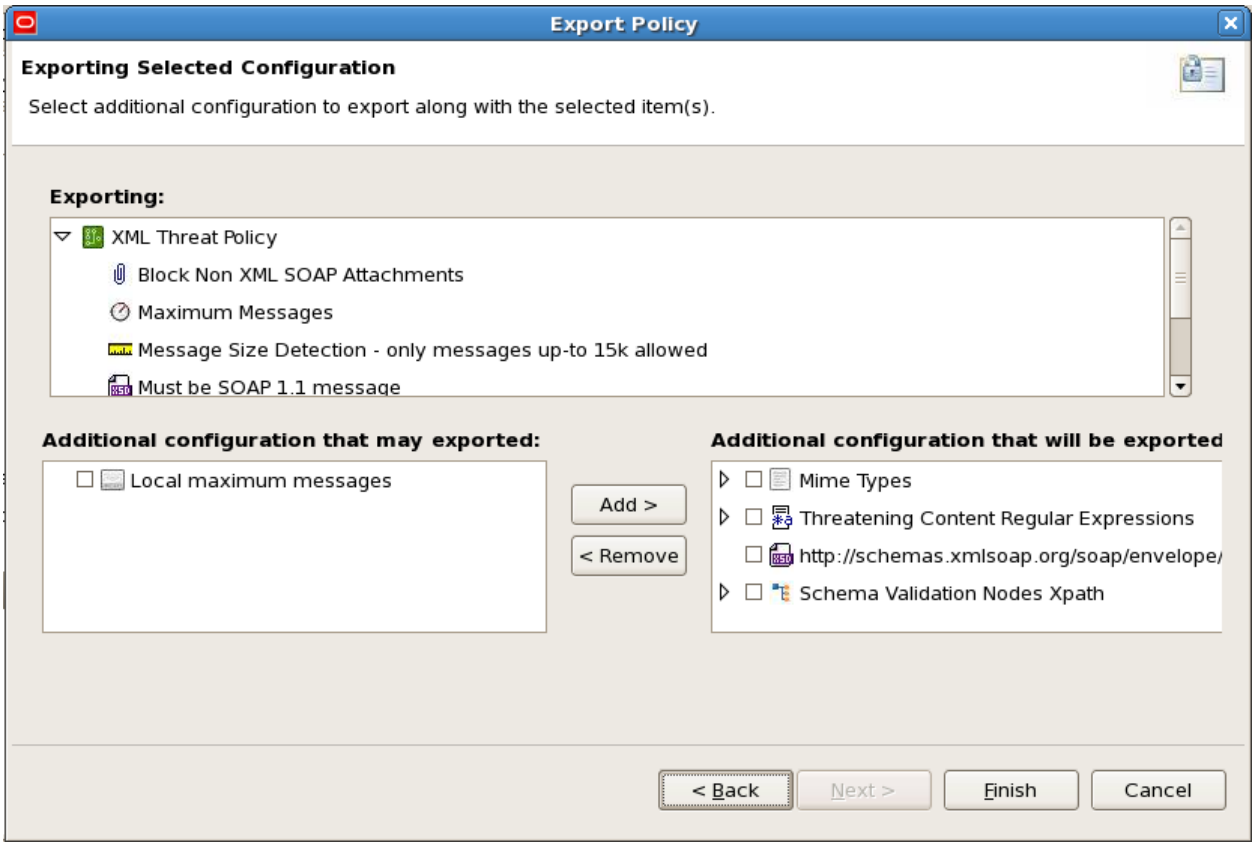
Tasks

1. In the Oracle Enterprise Gateway Dashboard, click Actions, and select Edit Active Configuration.
2. In the navigation frame, expand Policies > Policy Library
3. Right-click the XML Threat Policy node and select Export Policy. You should see a screen in the export wizard as shown below:



This is a read-only screen that displays the configuration items to be exported. The Exporting tree displays the selected tree node (in this case, XML Threat Policy), which is exported by default. “The following configuration items will also be exported” includes additional referenced items that are also exported by default along with the policy, such as MIME types, regular expressions, and schemas.

4. Click Next. You can refine the selection in this screen:



You can select optional configuration items for export. The “Additional configuration items that may be exported” tree on the left includes dependent items that are not exported by default (in this case, Local maximum messages, which is a local cache).

Note: When exporting a policy or policy container, by default, any policies referenced by the policy are included for export and displayed in the “Additional configuration that will be exported” list.

5. You can add this local cache for export by selecting it in the tree on the left, and clicking Add.
6. Click Finish.
7. In the upcoming dialog box, navigate to `labs/Lesson_06/`, and name the exported policy `XMLThreats.xml`. Click OK.
8. You can open a File Browser to verify that the policy is exported in the `XMLThreats.xml` file

Practices for Lesson 7: Fault Handling

Chapter 7

Practices for Lesson 7: Overview

Practices Overview

In these practices, you will view the fault message produced by OEG's default fault handler, and then replace it with a customized fault handler (already created for you) for your registered web service.

Practice 7-1: Testing the Service Using the Default Fault Handler

Overview

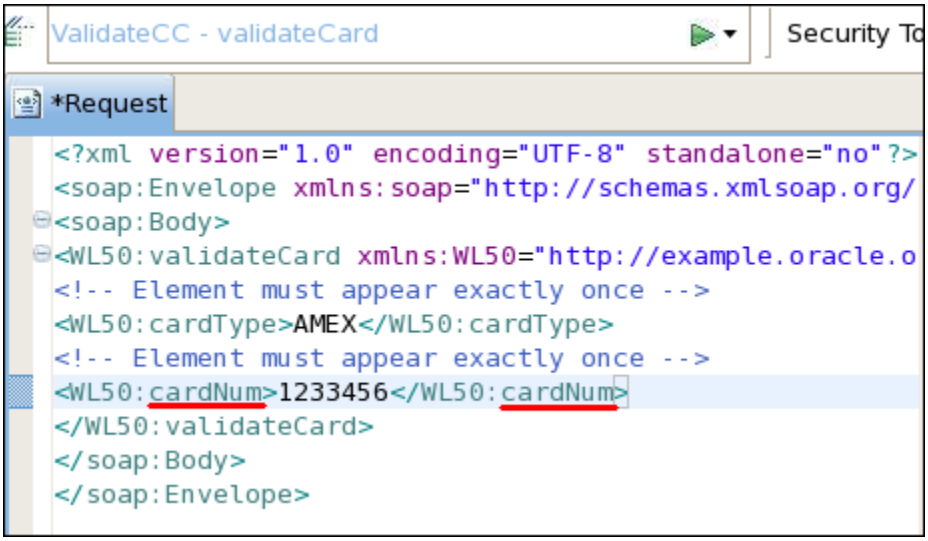
In this practice, you send a request with an invalid element name to the ValidateCC service, and see the default fault message response.

Assumptions

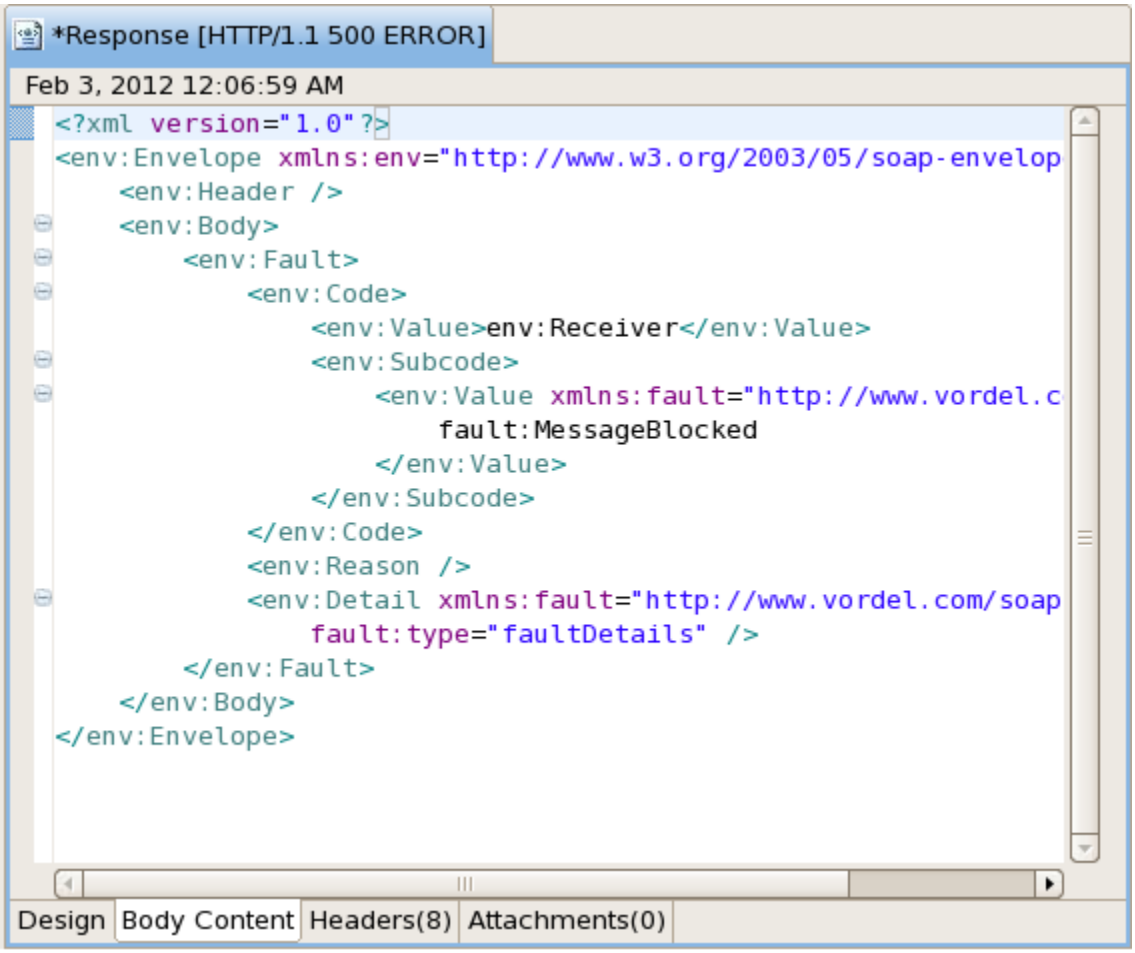
The Enterprise Gateway and WebLogic Admin server are up and running.

Tasks

1. Open Service Explorer, and select ValidateCC – validate Card request.
2. Change the request message to include the element <cardNum> instead of <cardNr>, so the request remains a valid XML (opening tag matches closing tag) document, but will fail the schema validation. The change is shown below:



3. Click Run. You should see a response message displayed with 500 error. The SOAP fault message you received is from the default fault handler.



4. You can see the blocked message in the Real time monitoring console. The Alert generated in this case is under the Audit and Alerts messages tab in Real time monitoring.

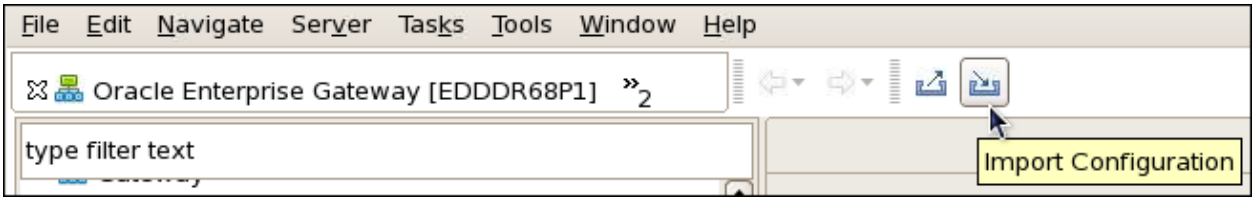
Practice 7-2: Importing and Viewing the Fault-Handling Policy

Overview

In this practice, you import and view some fault-handling policies that were created for you.

Tasks

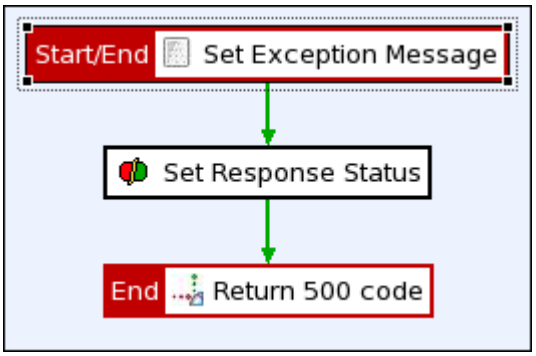
1. Open Policy Studio, click the Import Configuration button on top of the toolbar.



2. In the pop-up window, navigate to `labs` > `Lesson_07` folder, and select the `FaultHandlers.xml` policy.
3. If you see a warning message, click Yes to proceed.
4. In the Enter Passphrase dialog box, click Ok.
5. Review the content in the Import Configuration window, and click OK.

   After the policy is imported successfully, you should see a policy container named Fault Handlers displayed under the Policies node.
6. The policy container contains three policies as shown below:

7. Click the Global Exception Handler policy, and view its circuit on the canvas.



You can open each filter to view its configuration.

- The Set Exception Message filter returns the client a message with a specific error code and error message.
- The Set Response Status filter is used to explicitly set the response status of a call. This status is then recorded as a message metric for use in reporting. This filter is primarily used in cases where the fault handler for a policy is actually a Policy Shortcut.

8. The other two policies are specific handlers, which set the error code and error message

   Note: This practice uses only the Global Exception Handler. The other two will be used in the subsequent labs.

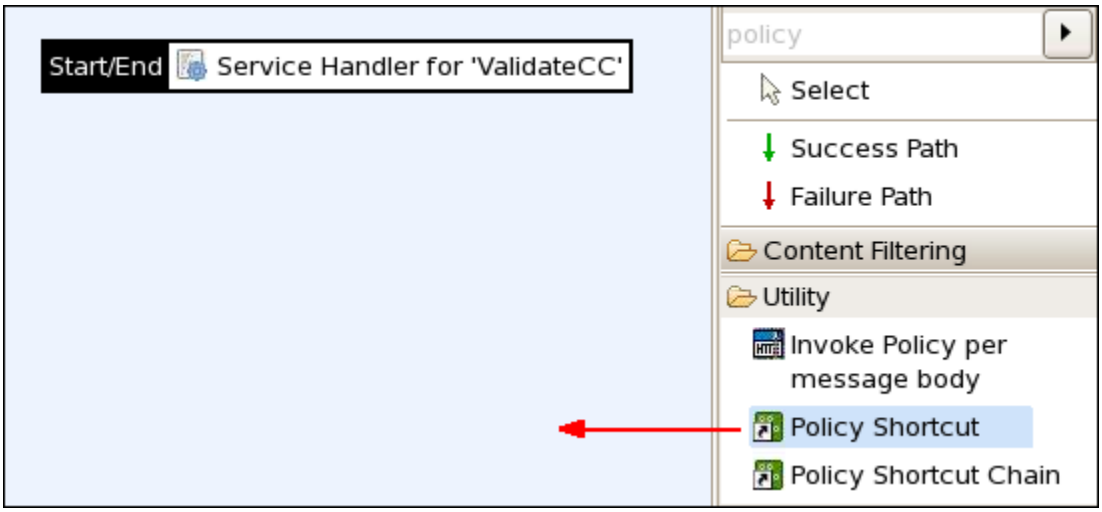
Practice 7-3: Adding Global Fault Handler

Overview

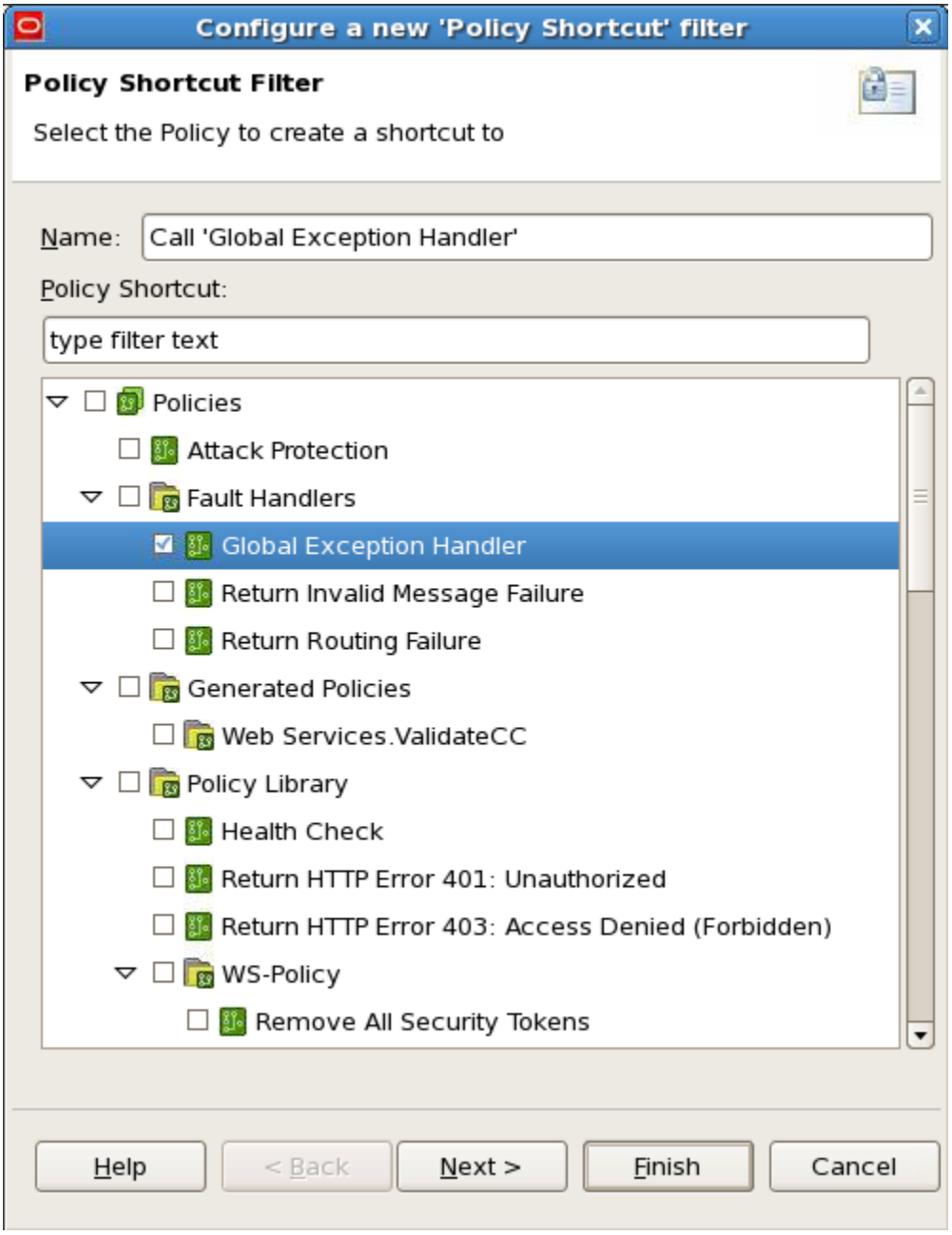
In this practice, you configure the ValidateCC web service to use this Global Exception Handler, a customized fault handler, to replace the default one.

Tasks

1. Navigate to the policy generated for the ValidateCC service under Policies > Generated Policies > Web Services.ValidatedCC
2. Click the `/validatecc/ValidateCCPort` policy to edit it.
3. In the filter palette to the right, search for the “Policy Shortcut” filter (in the Utility category)
4. Drag the “policy shortcut” on the policy editor canvas. The filter editor opens.

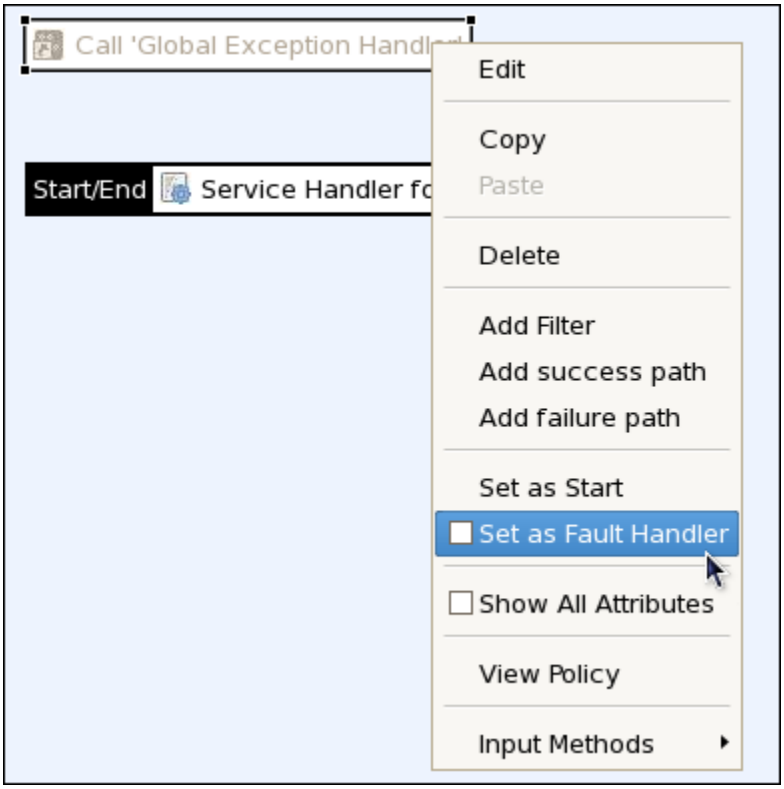


5. In the Configure a new ‘Policy Shortcut’ filter editor, select the Global Exception Handler policy.

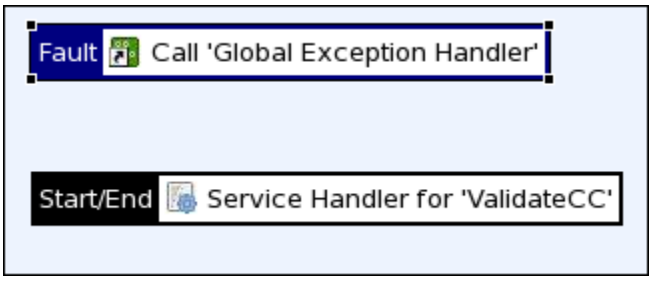


6. Click Finish. The policy shortcut filter appears as disabled on the editor canvas.

7. Right-click the policy shortcut filter and select Set as Fault Handler.

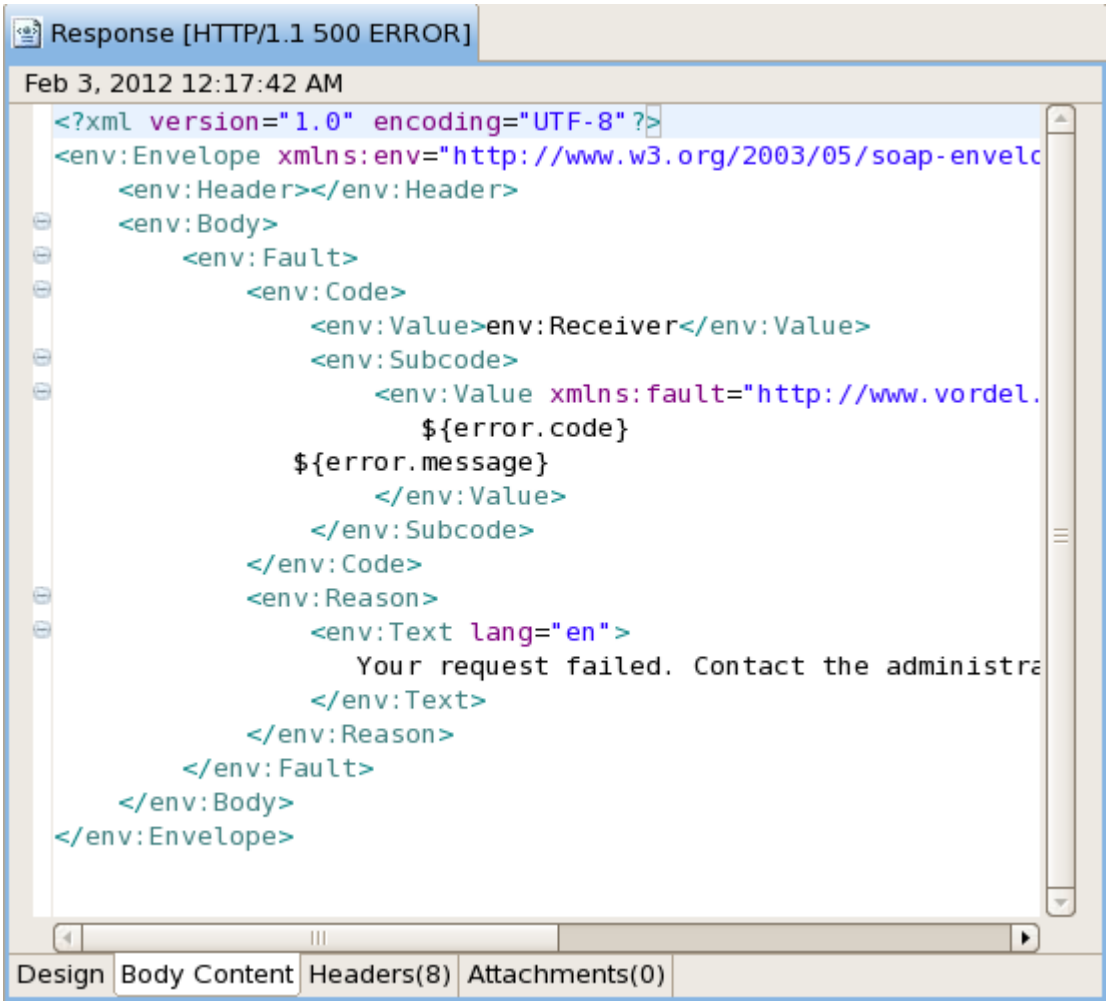


8. The policy shortcut now appears with a blue background, as shown in the image below.



9. Deploy the configuration by pressing F6.

10. Test the customized fault handler with the same wrong request and you will see the custom message defined in the global fault handler policy as a response.



Practices for Lesson 8: Blocking XML Threats

Chapter 8

Practices for Lesson 8: Overview

Practices Overview

A policy that scans for viruses and XML attacks has been created for you. In these practices, you apply the policy to the web service (ValidateCC), simulate the attacks by using Service Explorer, and view the Gateway to use the policy to protect the service from threats.

Practice 8-1: Applying Policy to Service

Overview

There is a prepared policy called Attack Protection that you need to import into Policy Studio. This policy checks:

- Incoming messages for viruses (using ClamAV)
- Incoming message complexity, to present XML “bomb attacks”
- Incoming message size
- Incoming messages for specific XML attacks, such as SQL injection

In this practice, you activate this policy by attaching it to the Service Handler that was generated when the service was registered.

Assumptions

The Enterprise Gateway is up and running.

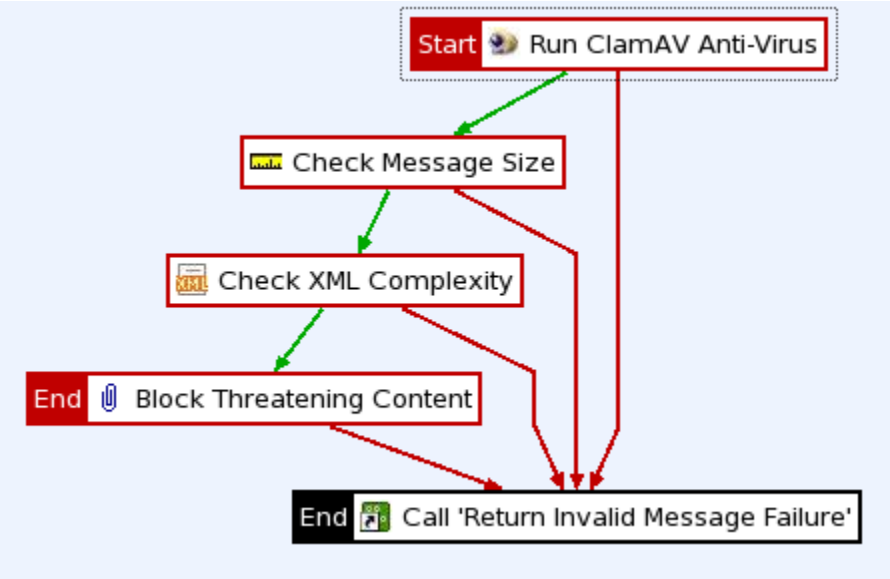
Tasks

Importing the policy

1. Open Policy Studio. Click the Import Configuration button on top of the toolbar.
2. In the pop-up window, navigate to the `labs` > `Lesson_08` folder, and select the `AttackProtectionPolicy.xml` policy.
3. If you see a warning message, click Yes to proceed.
4. In the Enter Passphrase dialog box, click Ok.
5. Review the content in the Import Configuration window, and click OK.

   After the policy is imported successfully, you should see a policy named Attack Protection listed under the Policies node.

6. You can open the policy and view its circuit.

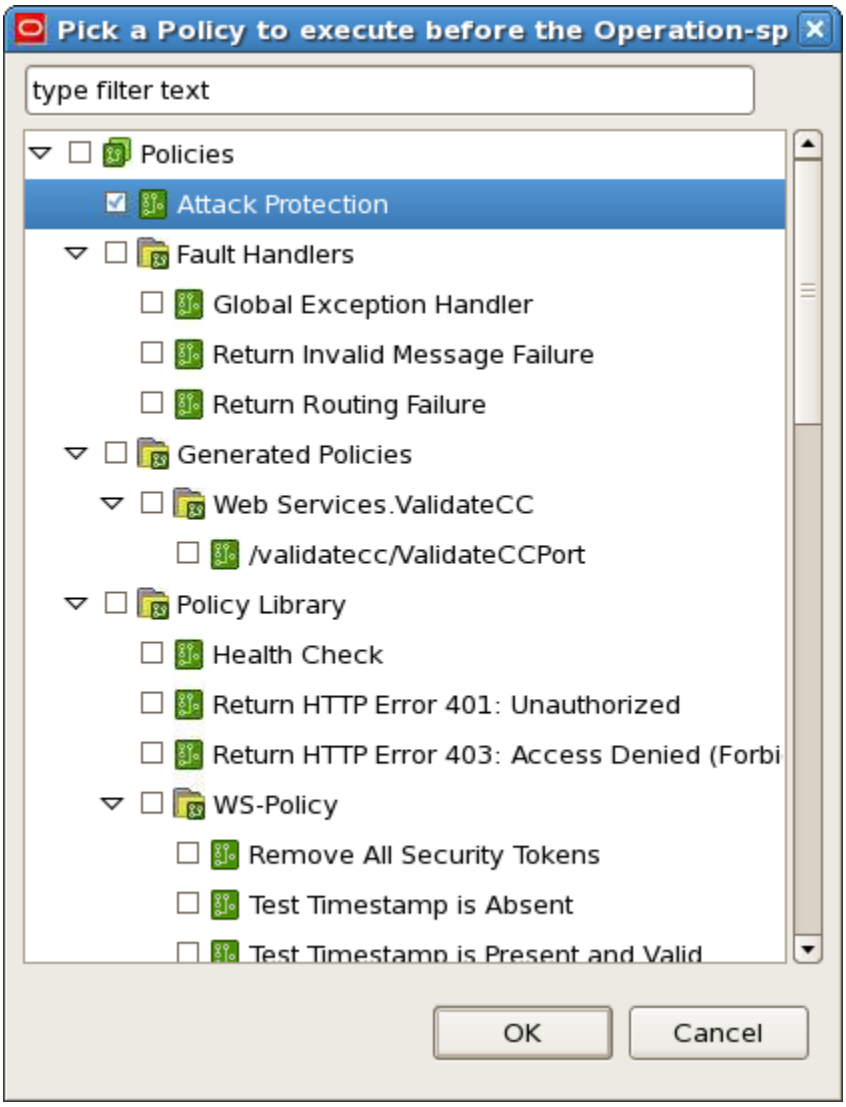


You can open each filter (double-click or right-click, selecting Edit) to view its configuration. Note that if the request message fails to pass any of the checks, a custom error message containing the error code and error message will be returned. Here, a Policy Shortcut filter is used to call a Fault Handler policy “Return Invalid Message Failure,” which was imported in the previous practice.

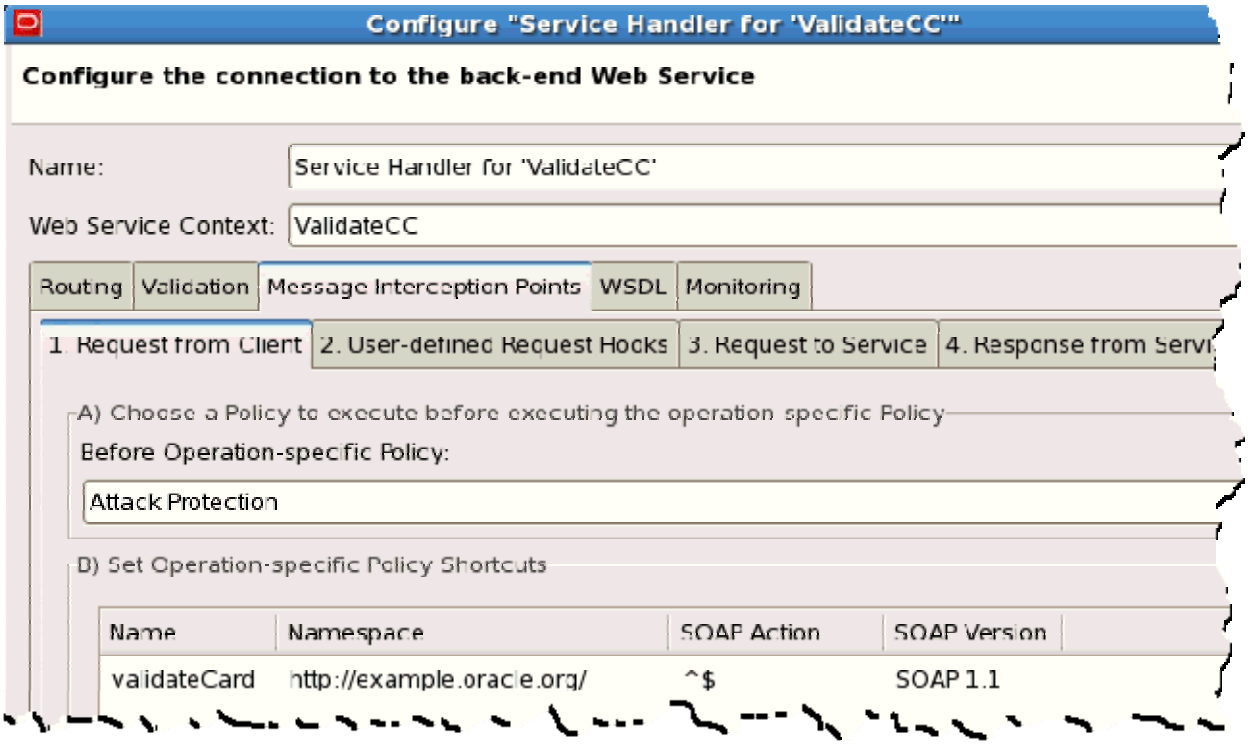
Attaching the policy to the Service Handler

1. Navigate to the policy generated for the ValidateCC service under Policies > Generated Policies > Web Services.ValidatedCC.
2. Edit the Service Handler for the ValidateCC filter by double-clicking it (or choosing right-click, Edit).
3. In the Configures “Service Handler for ‘ValidateCC’” window, select the Message Interception Points tab.
4. You want the Attack Detection policy to be the first policy to be called when a request arrives; therefore, you need to apply the policy when the Gateway receives a request from the client (tab 1), and before executing operation-specific policies (section A).

5. Click the ... button to the far right and select the Attack Protection policy in the list.



6. Click OK. The Attack Protection policy appears in the field.



7. Click Finish.
8. Deploy the configuration by pressing F6.

Practice 8-2: Testing with Service Explorer

Overview

In this practice, you will use three different attacks to test the service:

- A virus (the test virus known as EICAR)
- An XML “bomb” attack (recursive expansion)
- A SQL injection attack

Note that EICAR is a tiny file to test virus detection software. It’s not really a virus.

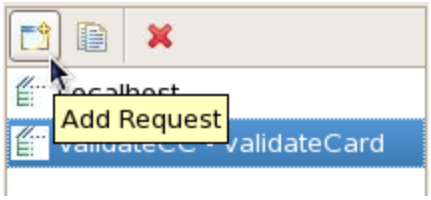
Assumptions

The Enterprise Gateway and the ClamAV service are up and running.

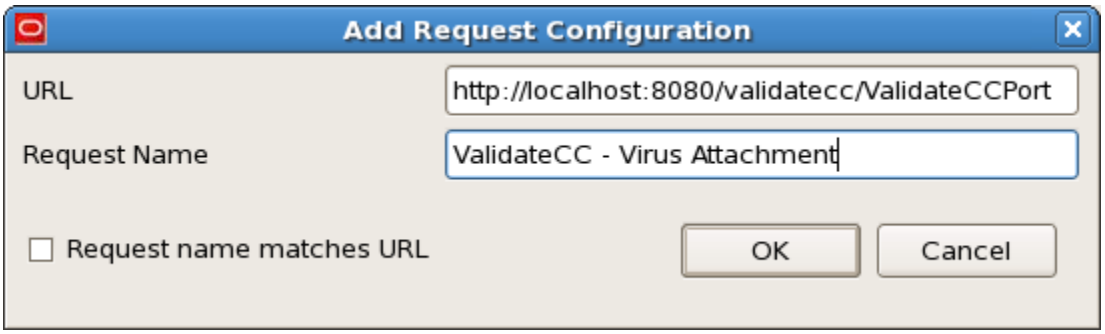
Tasks

Testing the virus attack

1. Open Service Explorer, click the inverted triangle in the toolbar, and select “Request Settings…” from the drop-down menu.
2. In the Request Settings window, click the “plus” button to add a new request.

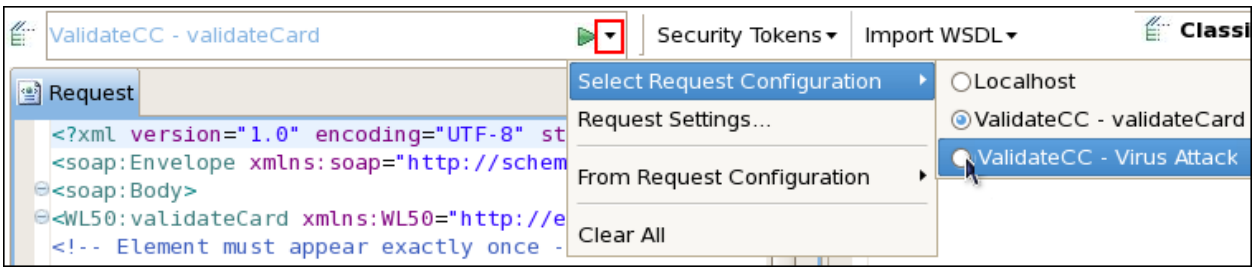


3. In the Add Request Configuration dialog box, configure the new request with the same URL of ValidateCC service, but a different request name:

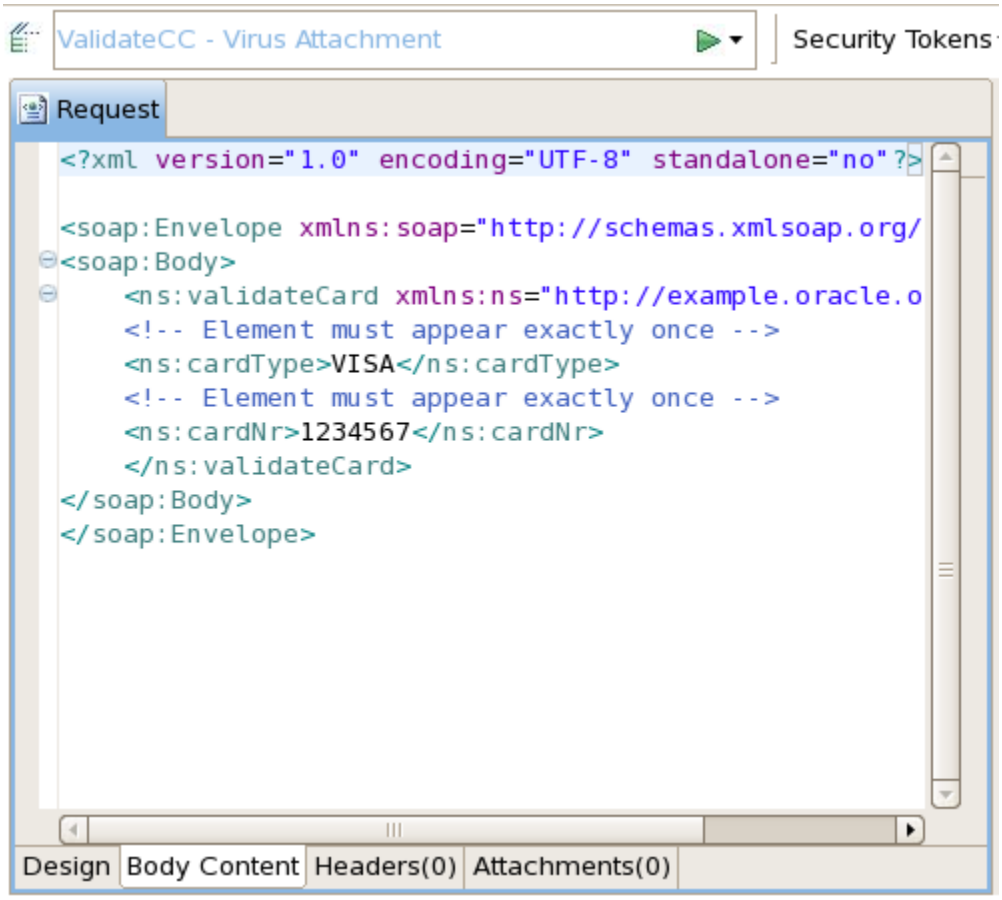


4. Click OK. The “ValidateCC - Virus Attachment” request is added to the request list.
5. Click Close to close the Request Settings window.

6. In Service Explorer, choose the “ValidateCC - Virus Attachment” request you just created. You should see no message in the Request tab.

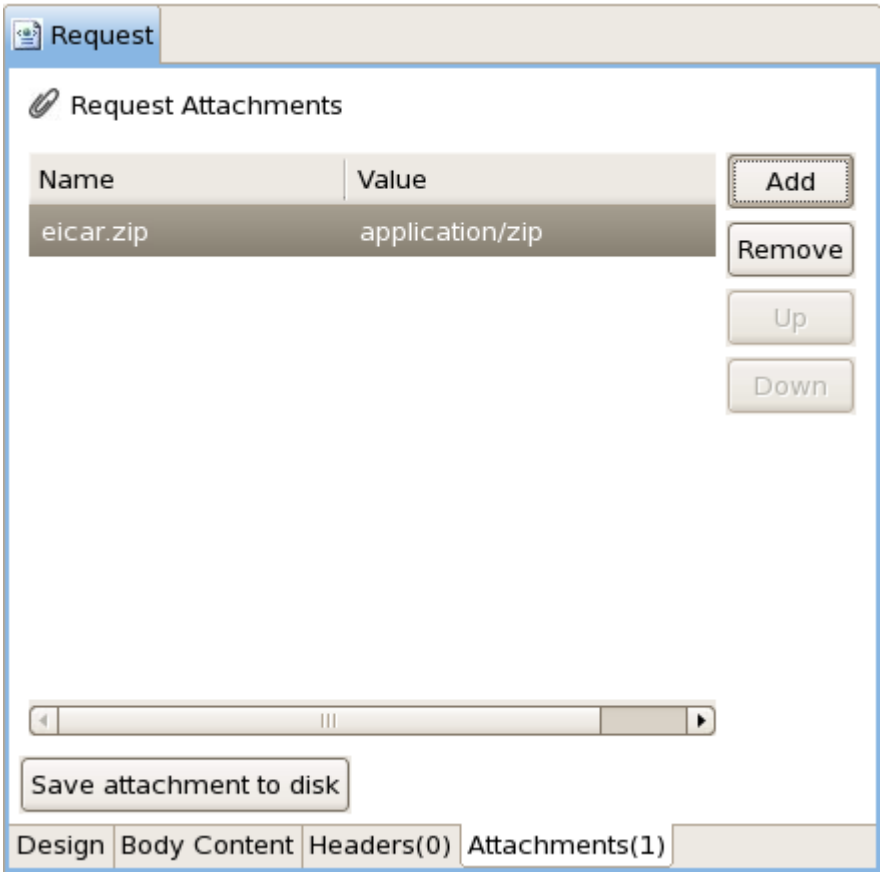


7. Select File > Load Request from the main menu.
8. Select the `VirusAttachmentRequest.xml` file from the `labs` > `Lesson_08` folder to load the request. You should see message content resembling the following screenshot displayed on the Request tab.



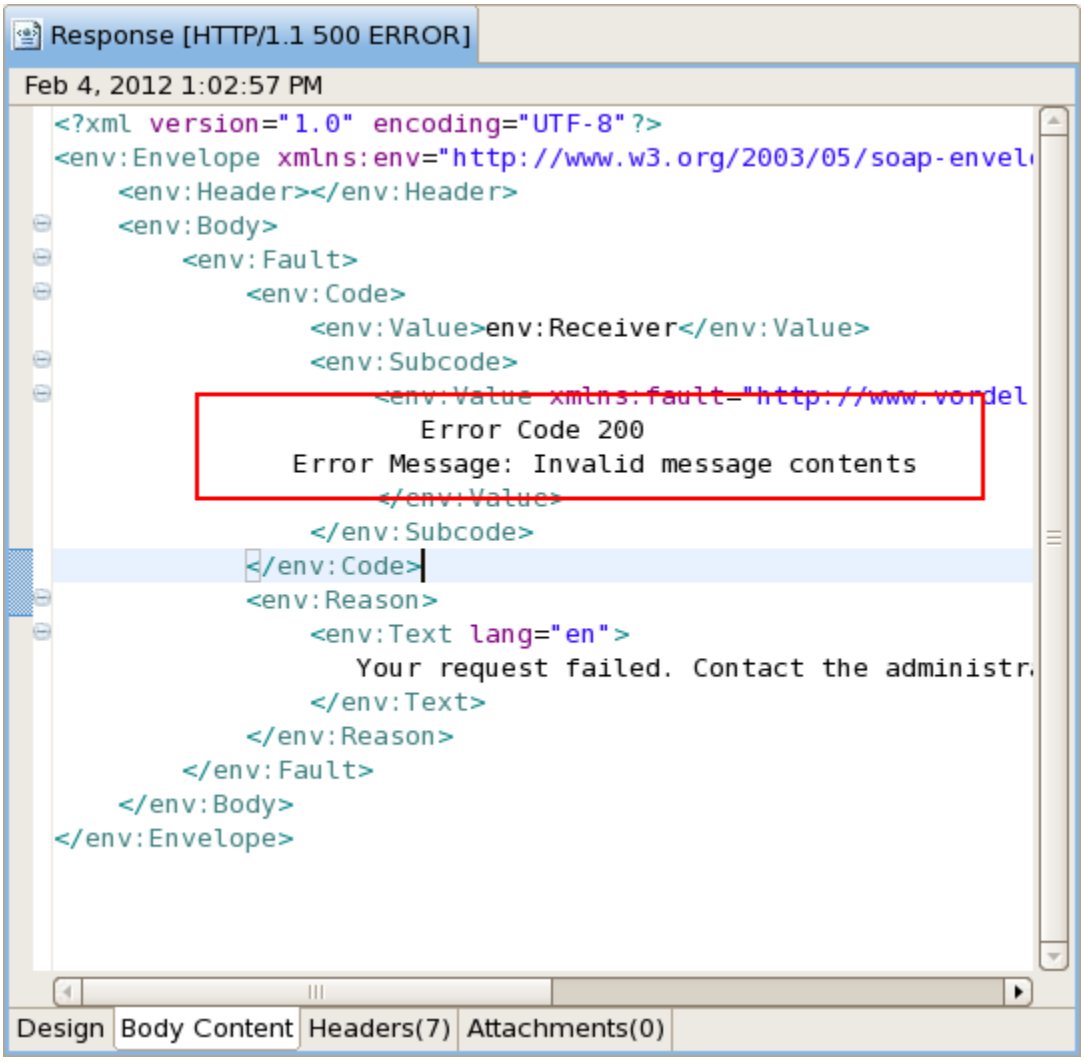
9. To attach the virus test file, perform the following steps:
   a. Click the Attachments tab at the bottom of the left pane.
   b. On the Request Attachments tab, click Add.

c. Select the `eicar.zip` file from the `labs` > `Lesson_08` folder to add it as the attachment. You should see the file displayed on the Request Attachment tab.



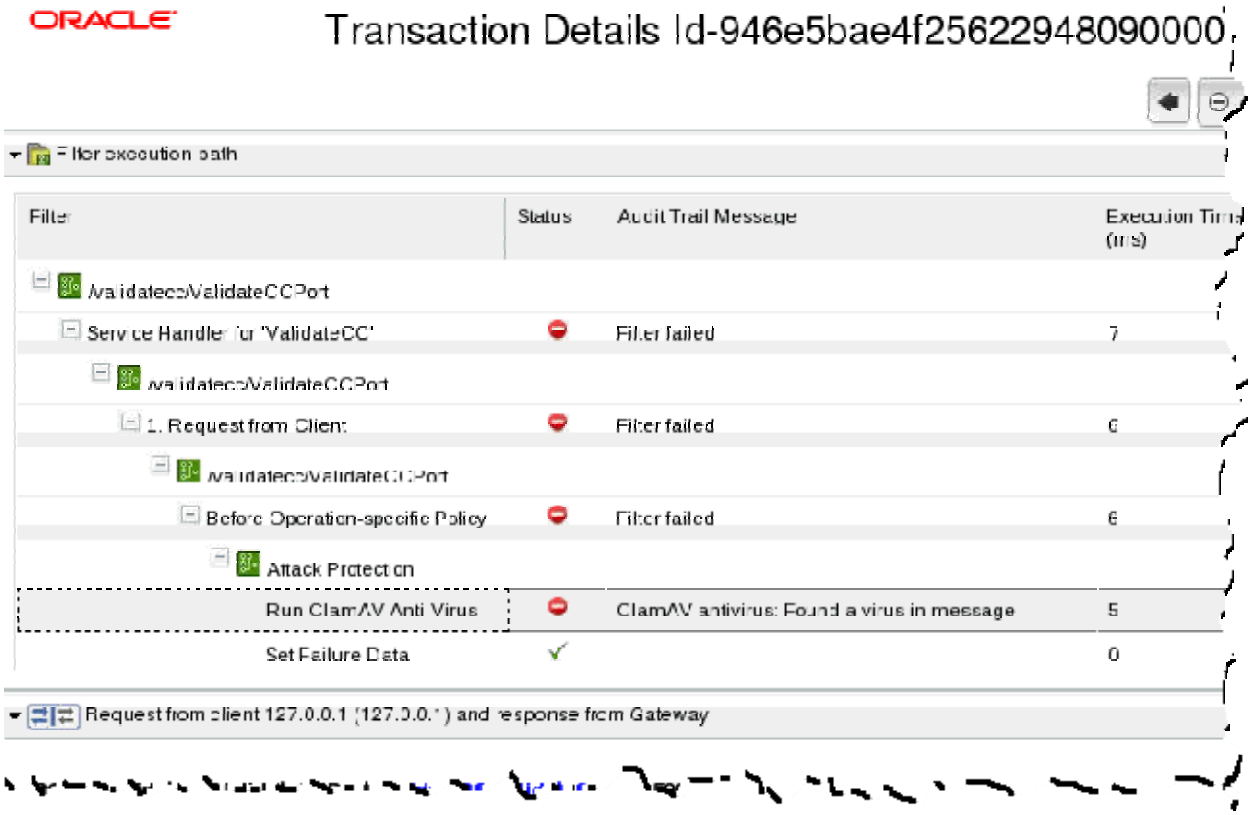
Note: EICAR is a special file used to test antivirus software.

10. Click the green “Play” button to run the request. You should get a 500 error response with a custom message as shown below:



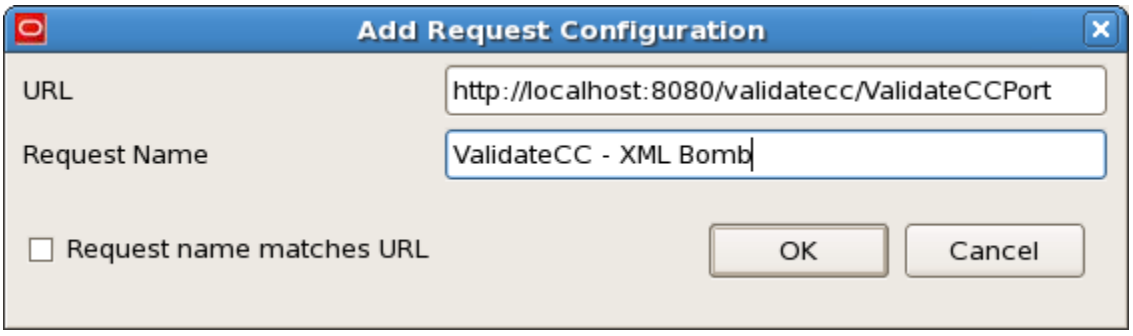
11. View the Real time monitoring and Audit Messages to check that the ClamAV anti-virus stopped the message.
    a. Open a Firefox browser, and go to the Gateway Management home page: http://localhost:8090/
    b. Click the “Traffic Monitor” link. The most recent request is listed on the top of the traffic log.

c. Click the request to view the transaction details, such as filter execution path, trace, and so on.



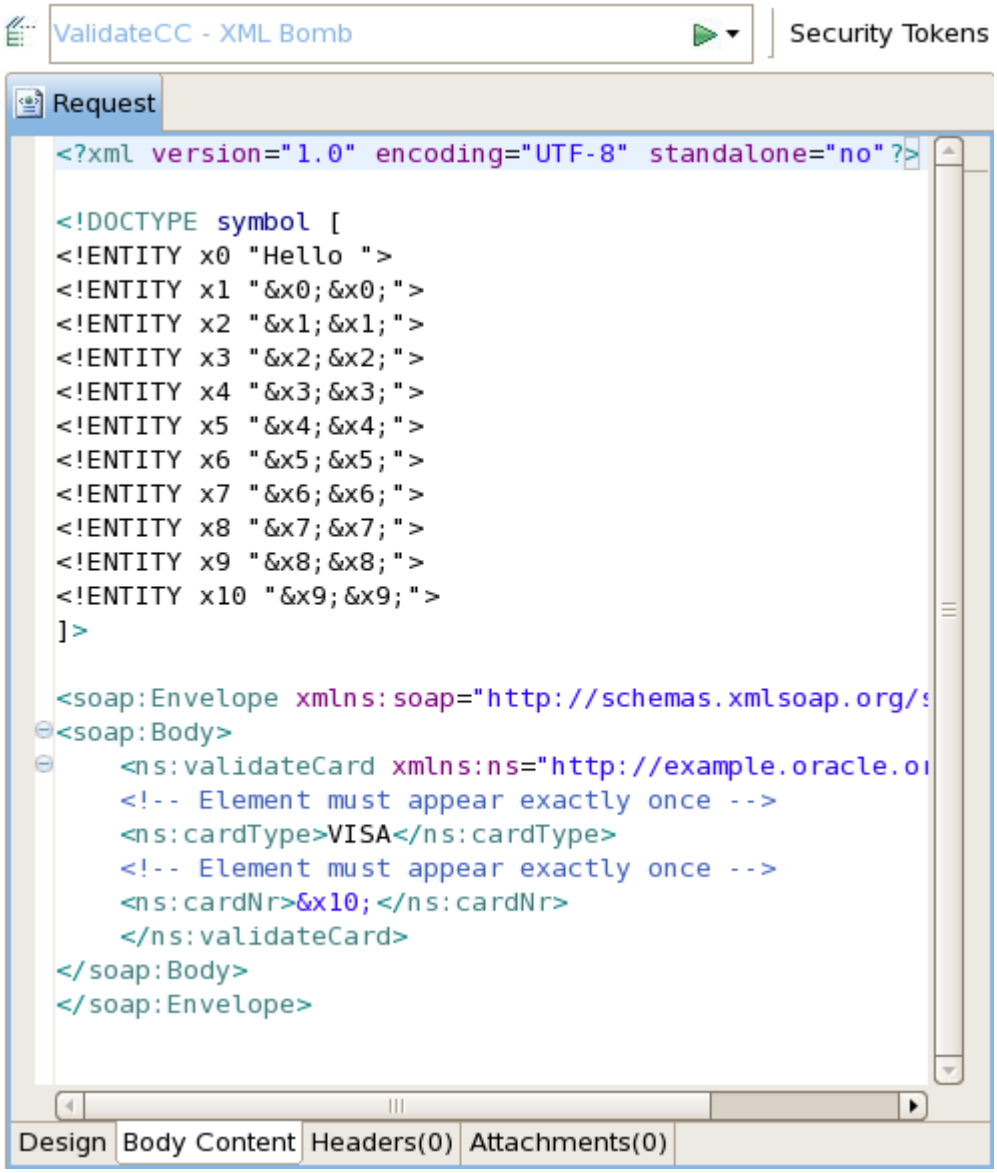
Testing the XML bomb attack

1. In Service Explorer, click the inverted triangle in the toolbar, and select “Request Settings…” from the drop-down menu.
2. In the Request Settings window, click the “plus” button to add a new request.
3. In the Add Request Configuration dialog box, configure the new request with the same URL of ValidateCC service, but a different request name: ValidateCC – XML Bomb:



4. Click OK. The “ValidateCC - XML Bomb” request is added to the request list.
5. Click Close to close the Request Settings window.
6. In Service Explorer, choose the “ValidateCC - XML Bomb” request you just created.
7. In the File menu, select Load Request.

8. Select the `XMLBombRequest.xml` file from the `labs` > `Lesson_08` folder to load the request. You should see the message content displayed on the Request tab. The request sends an XML recursive expansion attack.

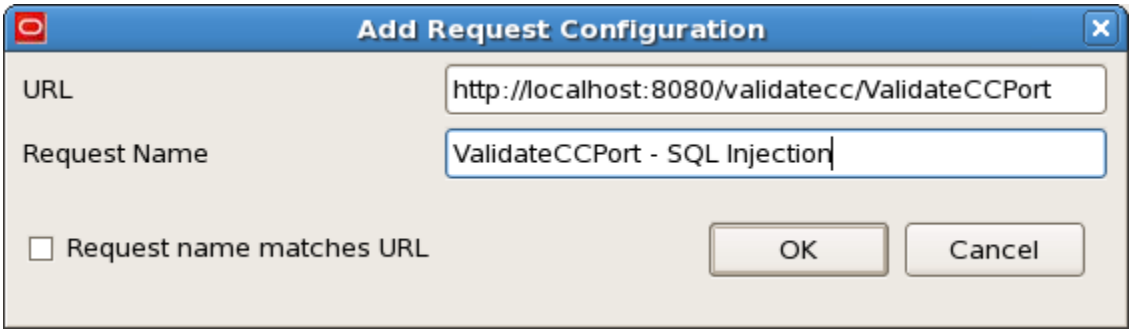


9. Click the green “Play” button to run the request. You should get a 500 error response message showing that the request is blocked.
10. View the Real time monitoring and Audit Messages. You will see that this attack is actually blocked by the XML parser itself.

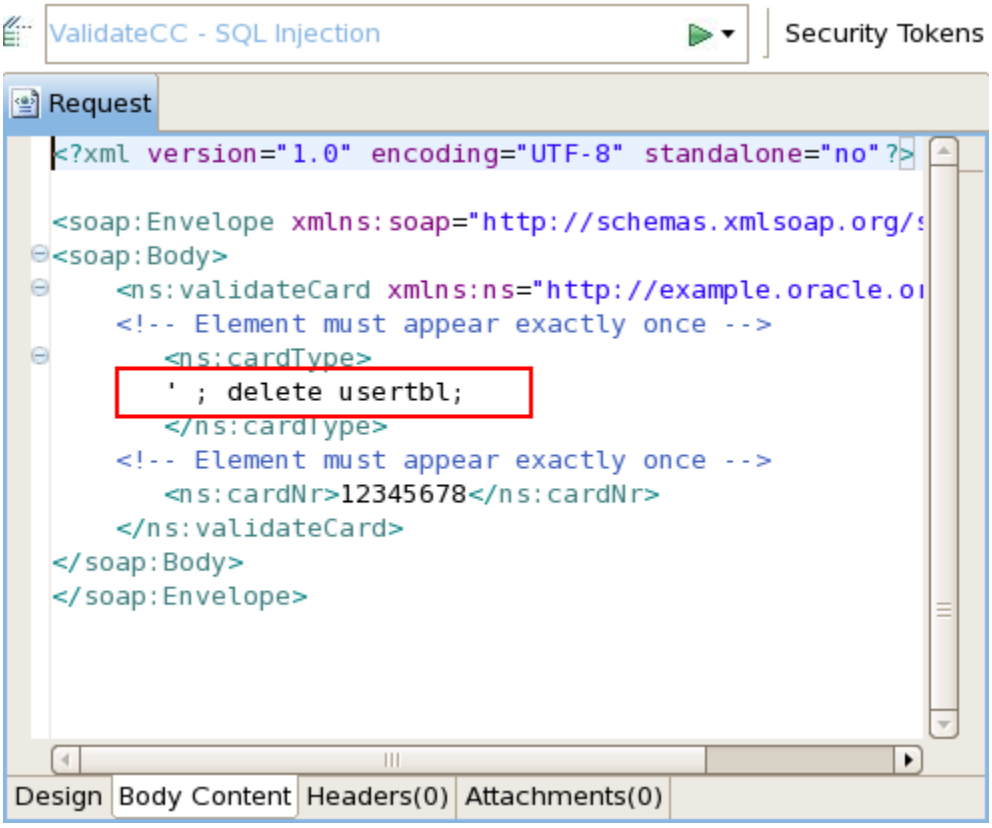
Testing the SQL injection attack

1. In Service Explorer, click the inverted triangle in the toolbar, and select “Request Settings…” from the drop-down menu.
2. In the Request Settings window, click the “plus” button to add a new request.

3. In the Add Request Configuration dialog box, configure the new request with the same URL of ValidateCC service, but a different request name: ValidateCC – SQL Injection:



4. Click OK. The "ValidateCC - SQL Injection" request is added to the request list.
5. Click Close to close the Request Settings window,
6. In Service Explorer, choose the "ValidateCC - SQL Injection" request you just created.
7. In the File menu, select Load Request.
8. Select the `SQLInjectionRequest.xml` file from the `labs` > `Lesson_08` folder to load the request. You should see the message content displayed in the Request tab. The request includes a SQL injection that attempts to delete a table.



9. Click the green "Play" button to run the request. You should get a 500 error response message showing that the request is blocked.
10. View the Real time monitoring and Audit Messages.

Practice 8-3: Protecting REST Service

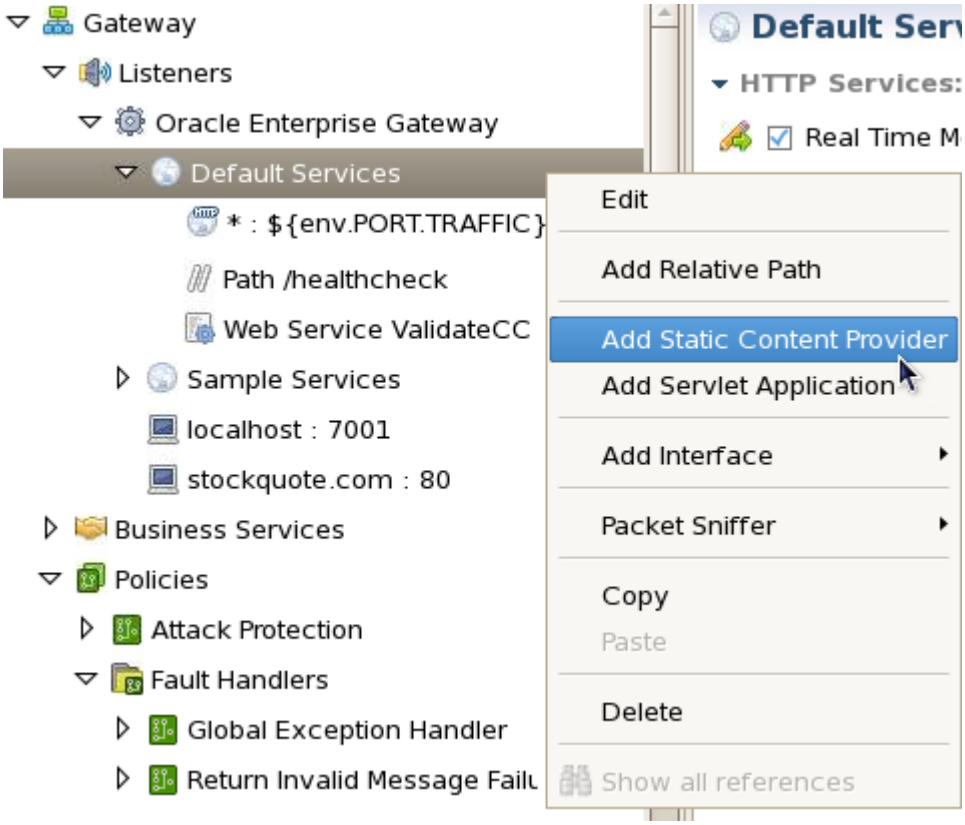
Overview

In this practice, you will apply a policy to a REST service in a browser. The policy validates the content of a parameter in the query string.

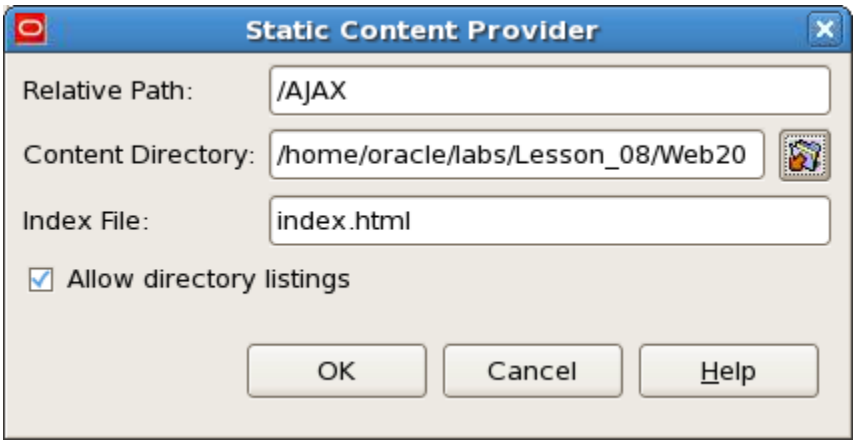
Assumptions

Tasks

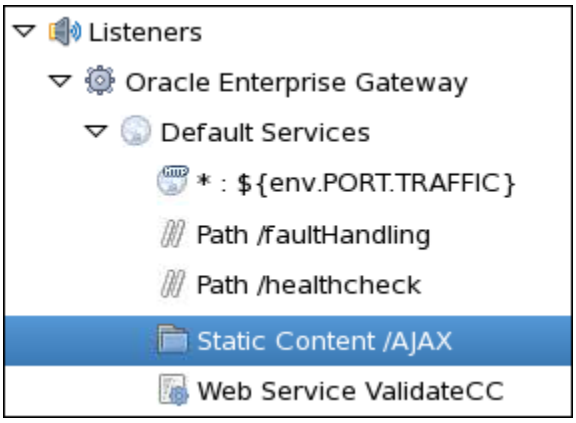
1. Open Policy Studio and navigate to Listeners > Oracle Enterprise Gateway > Default Service.
2. Delete the / relative path under the Default Services.
3. Add a “Static Content Provider” to serve out the AJAX page, as shown below.



4. In the Static Content Provider editor, configure the following two fields:
   - Relative Path: Enter the path `/AJAX`
   - Content Directory: Enter or browse to the `/home/oracle/labs/Lesson_08/Web20` directory

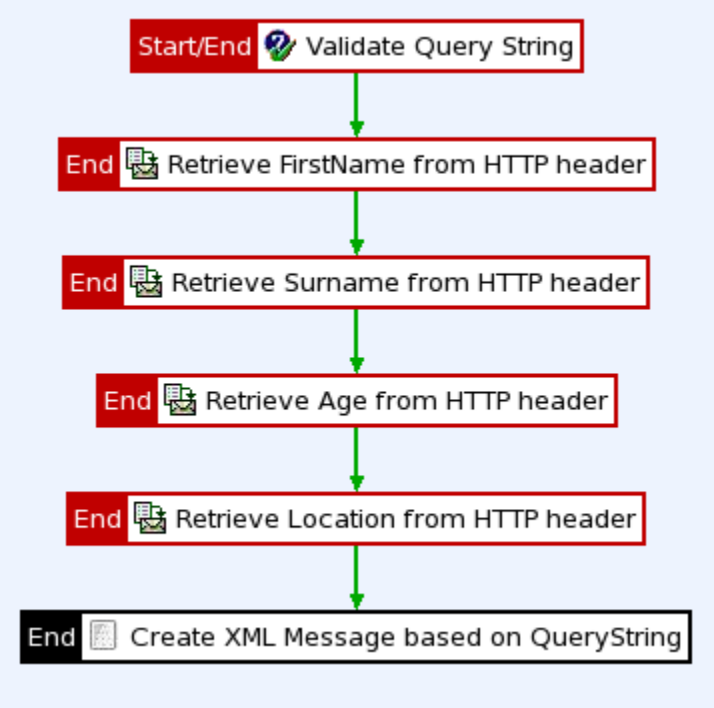


5. Click OK. The path is added under Default Services.



6. Import the `Web2Policy.xml` policy located in the `/home/oracle/labs/Lesson_08/Web20` directory.

7. Click the Web20 policy and examine the circuit (by opening each filter).

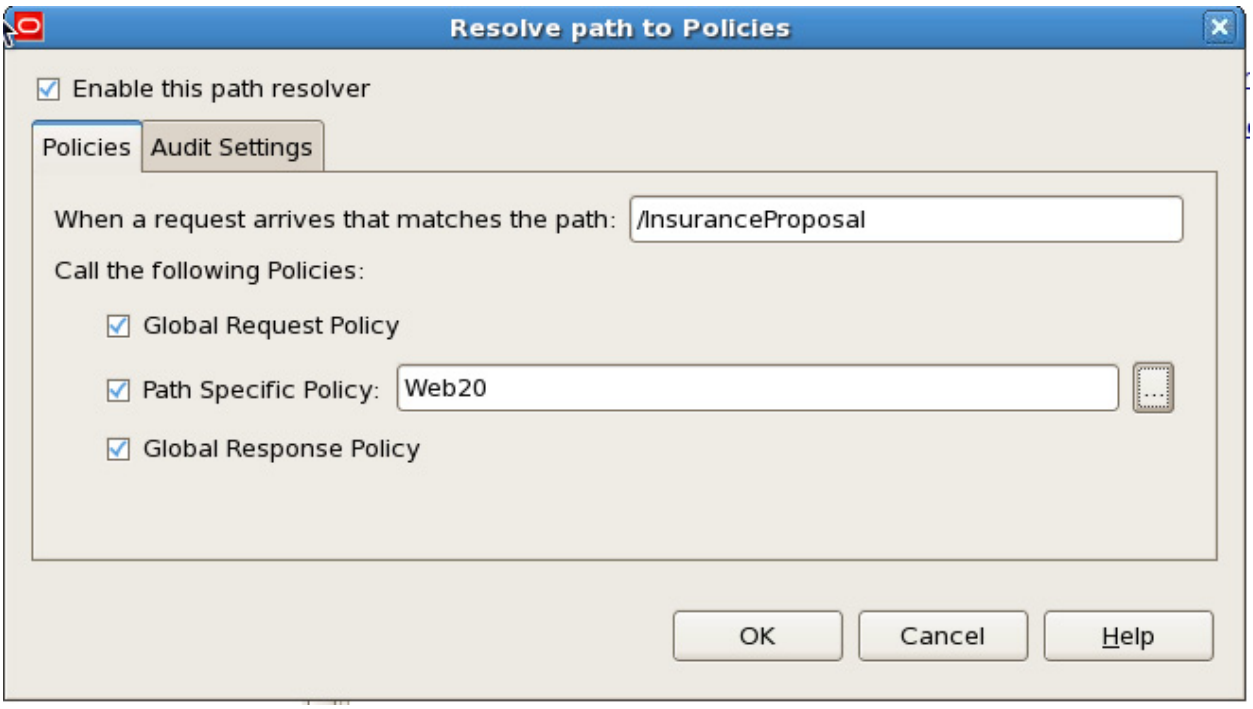


You will notice that it creates an XML message based on the REST parameters in the query string, and validates the data type of the “Age” parameter. You can see this XML message by directly accessing the REST web service by using a URL, for example:

http://localhost:8080/InsuranceProposal?FirstName=Joe&Surname=User&Age=25&Location=Boston

Note: If you want to access the above URL, deploy the configuration first.

8. Add a new relative path of `/InsuranceProposal` (under the Listeners > Oracle Enterprise Gateway > Default Services node), and map the Web20 policy to it.



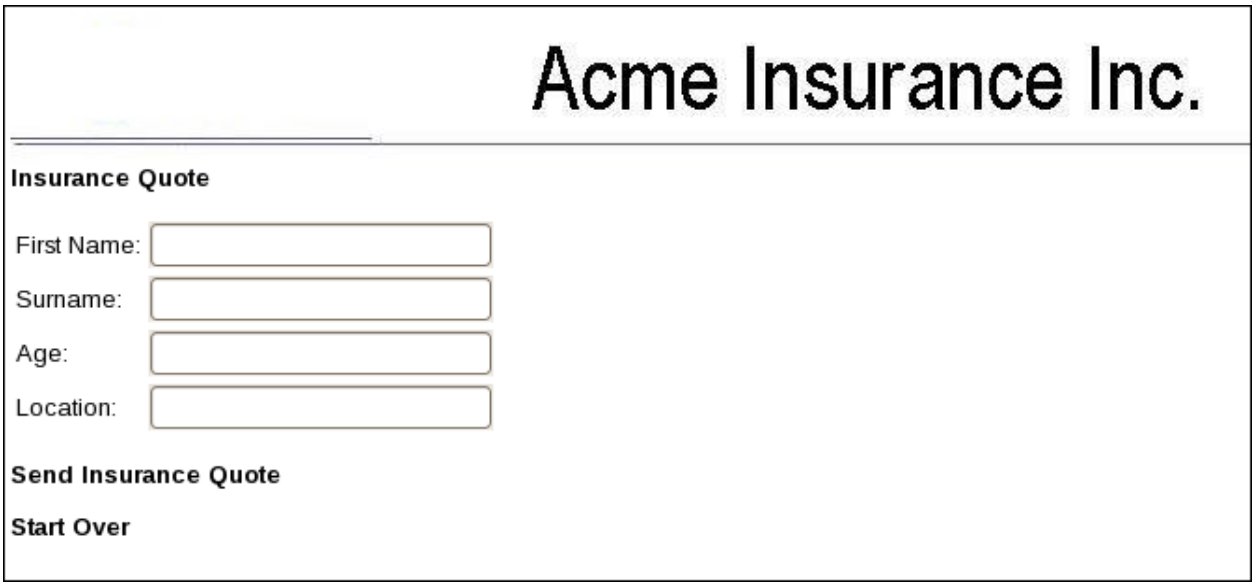
9. Deploy the configuration.

Testing

1. Open the browser, enter the following URL:

   http://localhost:8080/AJAX/AJAX-Page.html

   You should see a page resembles the following image:



2. To test the AJAX parameter validation, you can do the following:
   a. Enter alphabetic values in the First Name, Surname, and Location fields, and a numeric value in the Age field.
   b. Click Send Insurance Quote, and view the response message.
   c. Replace the numeric value with a non-numeric value in the Age field, click Send Insurance Quote again, and you should get the message-blocked response.

Practices for Lesson 9: Accelerating XML and Managing Traffic

Chapter 9

Practices for Lesson 9: Overview

Practices Overview

In these practices, you will learn how to:

- Cache response messages from the back-end service
- Manage incoming requests at the Gateway level or service level

Practice 9-1: Caching Response Messages from the Service

Overview

In this practice, you get the credit card validation data by using a caching policy (pre-created for you). The policy will try to get the data from the cache first. If it fails, the policy retrieves the data from the back-end service, and then caches the response.

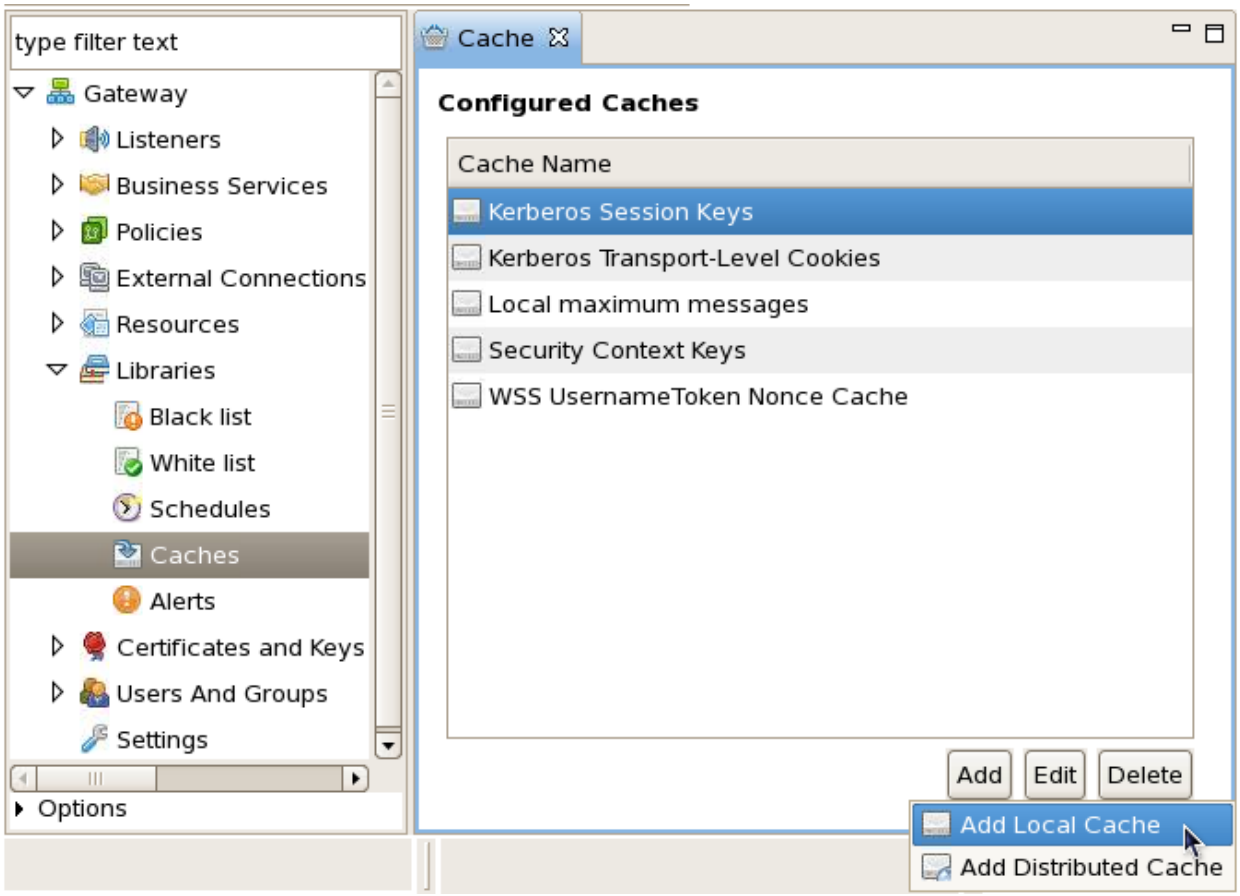
Assumptions

The Enterprise Gateway and WebLogic Admin server are up and running.

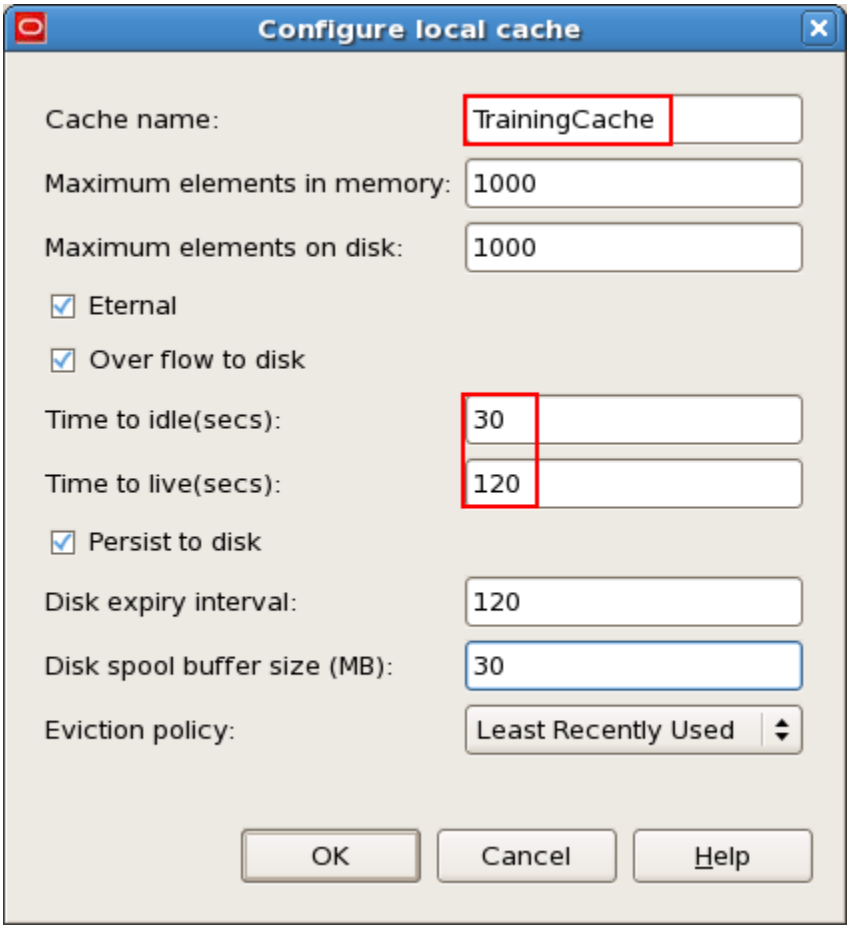
Tasks

Creating a local cache

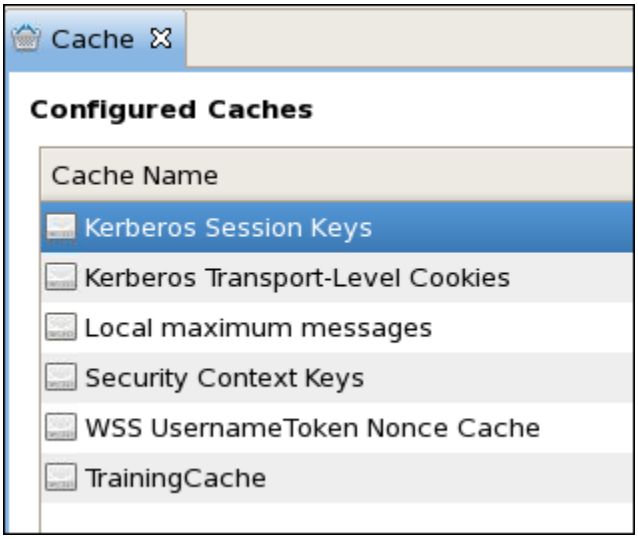
1. In Policy Studio, select the Libraries > Caches node, and click the Add button at the bottom right of the screen and select Add Local Cache.



2. Configure the fields on the Configure Local Cache dialog box as shown below:

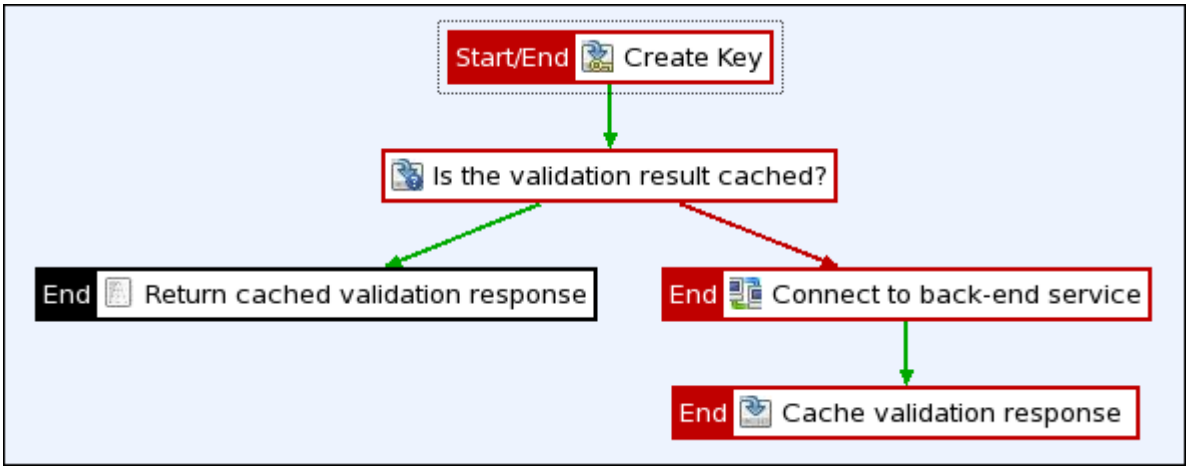


3. Click OK. The newly created cache is displayed in the Configured Caches list.



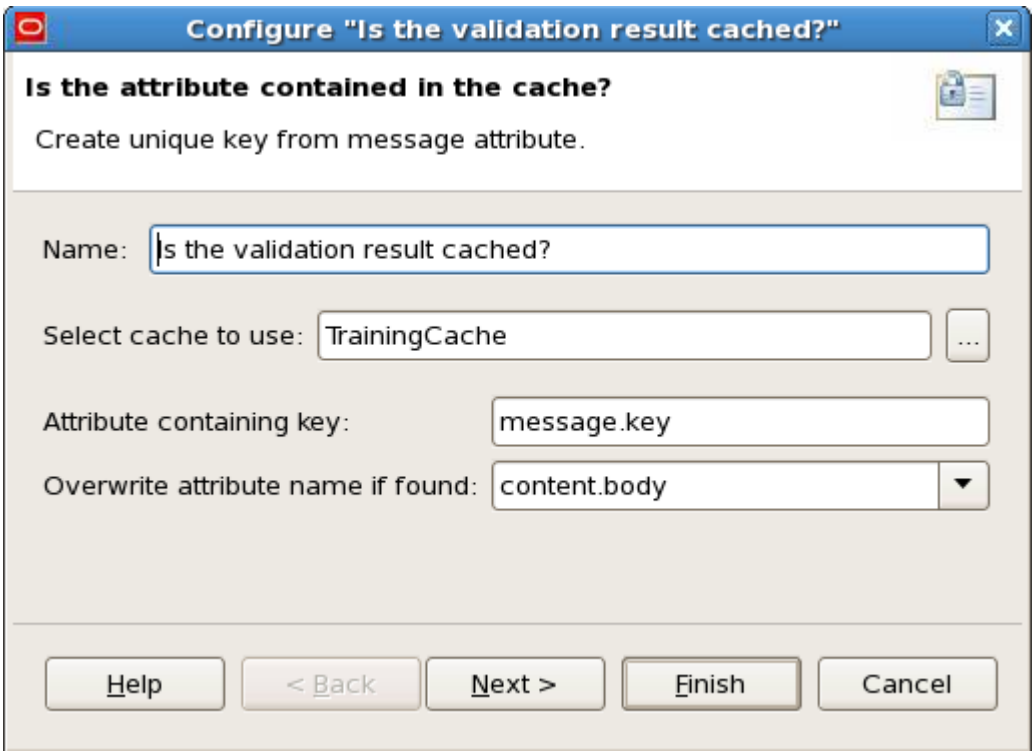
Importing and updating the caching policy

1. In Policy Studio, click the Import Configuration button on top of the toolbar.
2. In the pop-up window, in the `oracle` place, navigate to `labs` > `Lesson_09` folder, and select the `CachingPolicy.xml` file.
3. After the policy is imported successfully, you should see that a node named Caching is added under the Policies folder.
4. Select the policy to view its circuit in the policy editor, which resembles the following image:



5. Examine and modify the policy as described below:
   a. Open the “Create Key” filter (double-click or right-click, then select Edit). You can see the policy uses the content body as the unique key. Click Cancel.

b. Open the “Is the validation result cached?” filter. You should see TrainingCache is pre-populated in the “Select cache to use” field.



Note: If you see an error in the “Select cache to use” field, click the … (browsing) button next to the field, and select TrainingCache.

c. Open the “Return cached validation response” filter. If the validation data is retrieved from the cache, you will get a message indicating its cached data. Click Cancel.

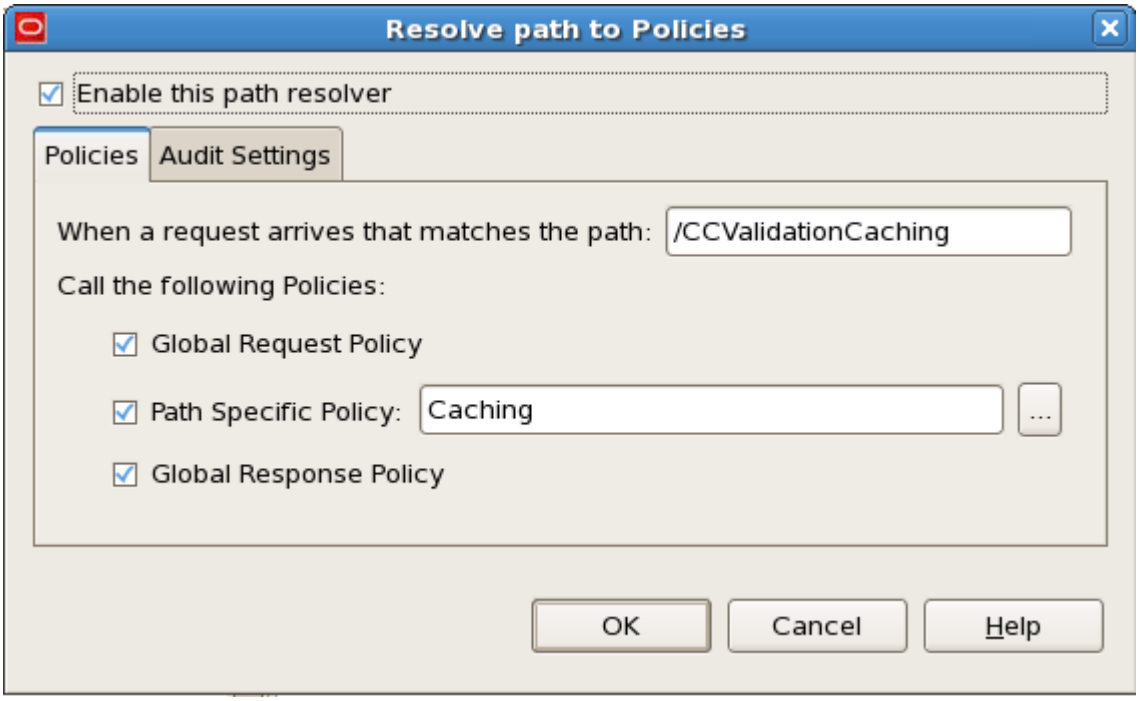
d. Open the “Connect to back-end service” filter. If the validation data can’t be retrieved from the cache, the Gateway will route the request to the back-end service to get the data. The URL of the service is defined here, which is http://localhost:7001/validatecc/ValidateCCPort.

e. Open the “Cache validation response” filter. After the Gateway gets the response, it will store it in the cache. Like step 8b, the TrainingCache value should be pre-populated in the “Select cache to use” field.

Mapping the policy to a Relative Path

1. Add a new Relative Path, and map the policy to it:
   a. Expand Listeners > Oracle Enterprise Gateway in the left tree pane, right-click the Default Services node, and select “Add Relative Path” from the context menu.
   b. In the Resolve path to Policies window, enter the new Relative Path as: `/CCValidationCaching`

c. Click the Browse button next to the Path Specific Policy field, and select the Caching policy.



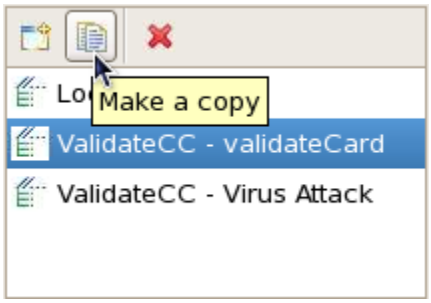
d. Click OK. You should see the /CCValidationCaching Relative Path is added under the Default Services node.

2. Deploy the configuration by using the Deploy button on the toolbar.

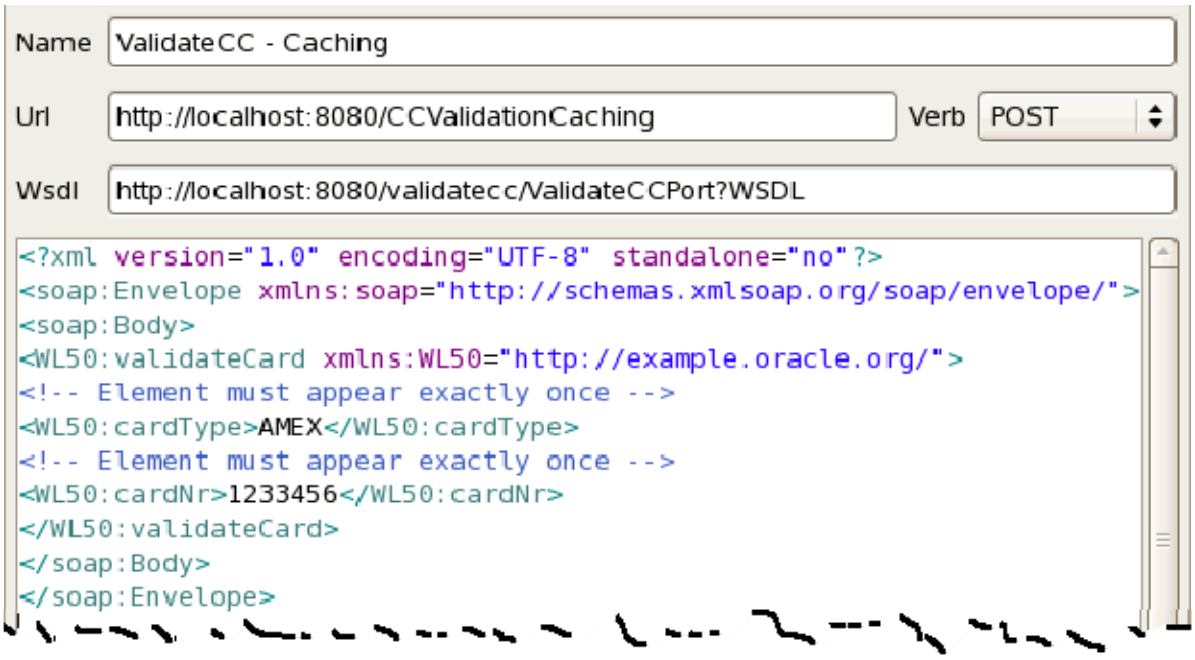
Testing the policy

You can use the same request message as the ValidatedCC – validateCard request to test this new policy.

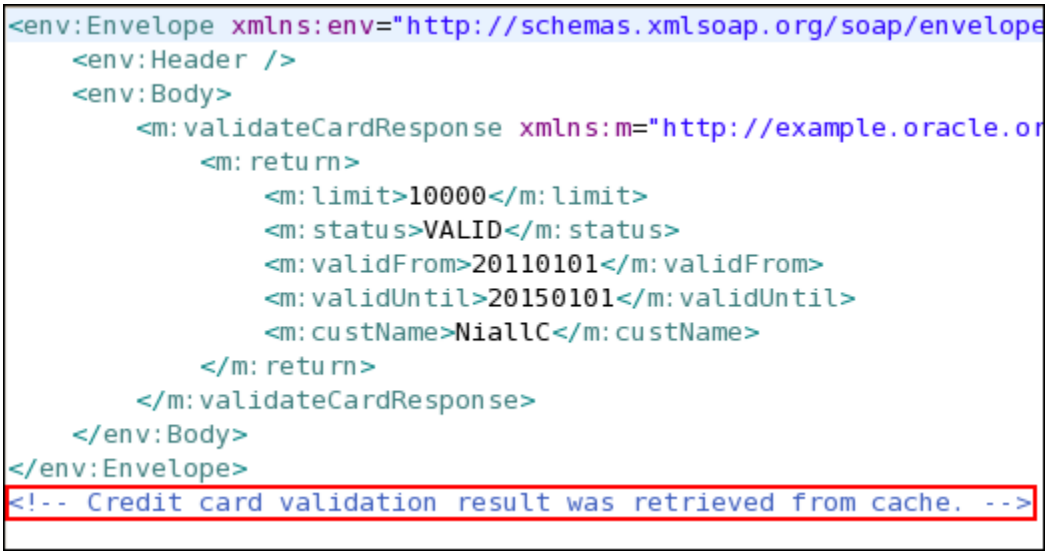
1. Open Service Explorer, click the inverted triangle in the toolbar, and select “Request Settings…” from the drop-down menu.
2. In the Request Settings window, select ValidateCC – validateCard, and then click the Make a Copy button on the toolbar.



3. A new request named Copy of ValidateCC – validateCard is displayed in the request list. Select the request, and make the following modification to the configuration in the right frame:
   - Change the name to `ValidateCC - Caching`
   - Change the URL to `http://localhost:8080/CCValidationCaching`



4. Click Run. You should see the same response message as you sent the request by using the ValidatedCC – validateCard request in the earlier practice.
5. You can open the browser and look at the Real time monitoring and Audit Messages.
6. Click the green "Play" button to send the request again. You should see the response from the cache as shown in the screenshot below:



7. Go to the browser, and look at the Real time monitoring and Audit Messages again.

Practice 9-2: Configuring Gateway-Wide (Global) Throttling

Overview

OEG limits inbound traffic by keeping track of requests either globally or by client. The incoming requests can be managed at the gateway level or at the service level. In this practice, you will apply request-throttling on a single gateway across all services, and see how to apply conditional throttling.

Assumptions

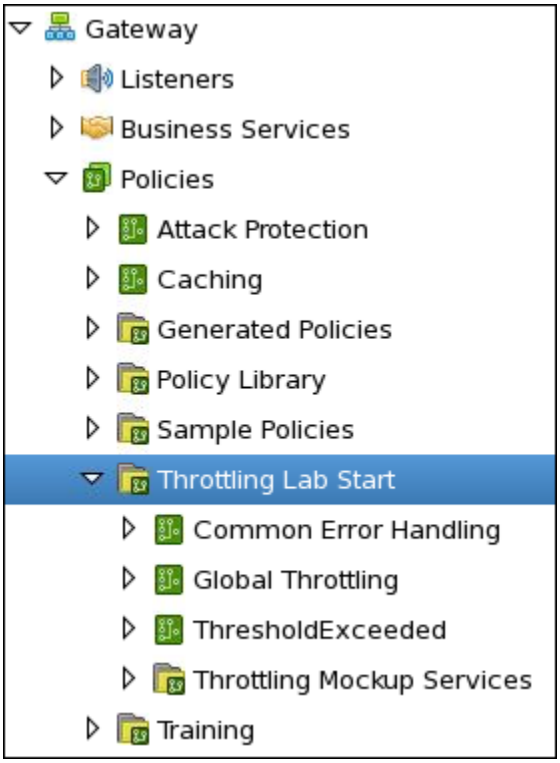
The Enterprise Gateway is up and running.

Tasks

Importing Startup Policies

1. Open Policy Studio and click the Import Configuration button on top of the toolbar.
2. In the pop-up window, navigate to `labs` > `Lesson_09` folder, and import the `ThrottlingLabStart.xml` file.

   After the policies are imported successfully, you should see that a policy container named "Throttling Lab Start" appears under the Policies node. Expand the node; you should see three policies and a subcontainer as shown below:

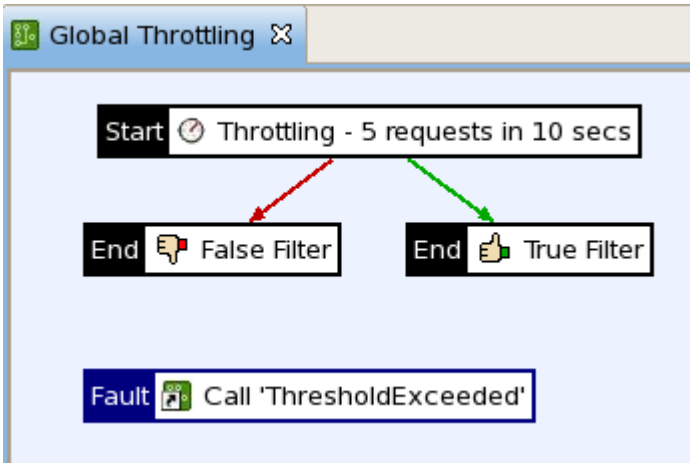


Setting the Gateway-Wide Throttling Policy

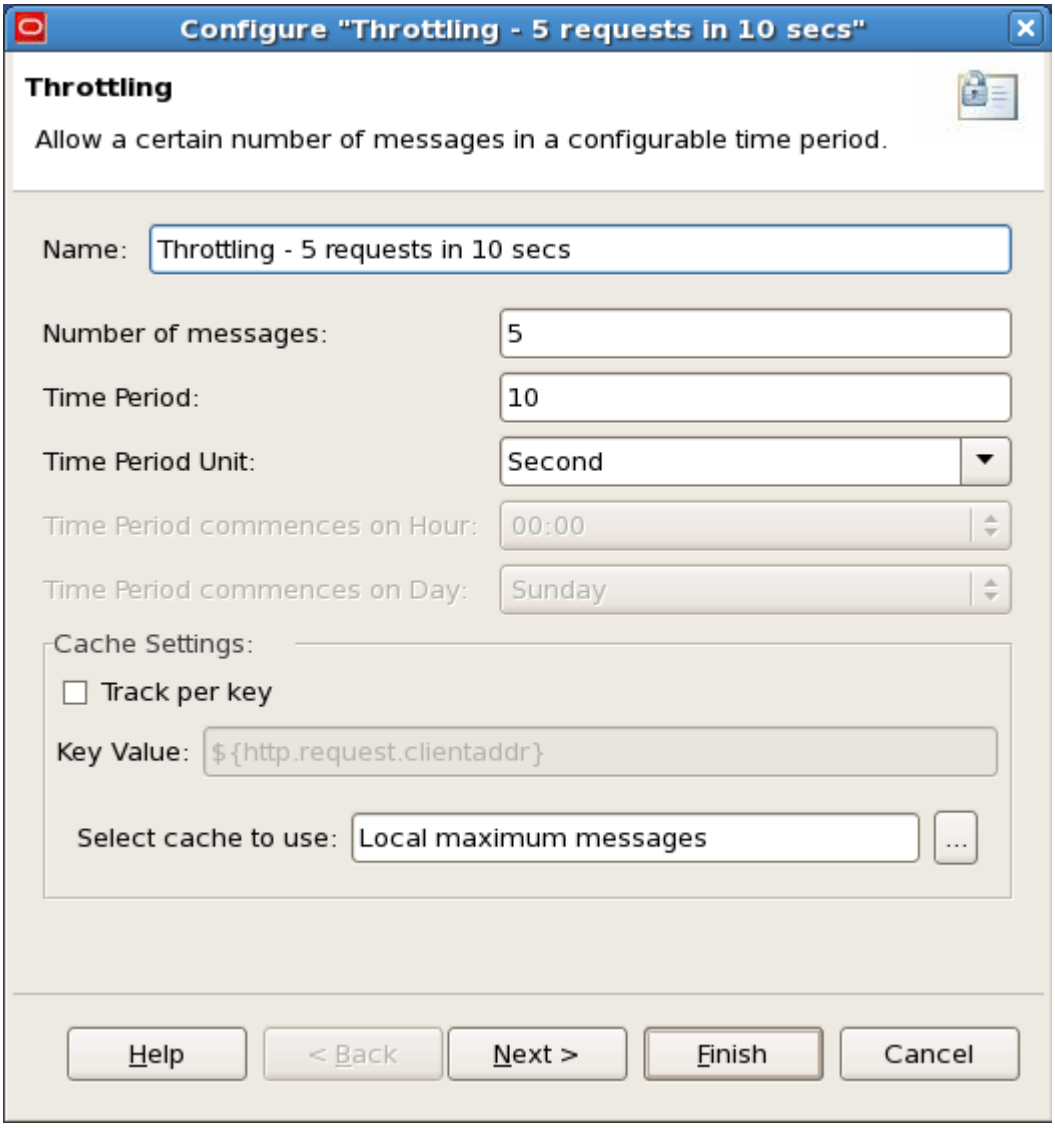
You can use a global throttling policy to restrict the total number of requests that are processed in a specified time period by the Gateway. It is typically:

- Used to protect all downstream services from DoS (rate-based) attacks
- Used as a capacity-planning aid and service-consumption monitor

3. Select the Global Throttling policy to view the circuit in the Policy editor:



a. Open the throttling filter, as shown below:



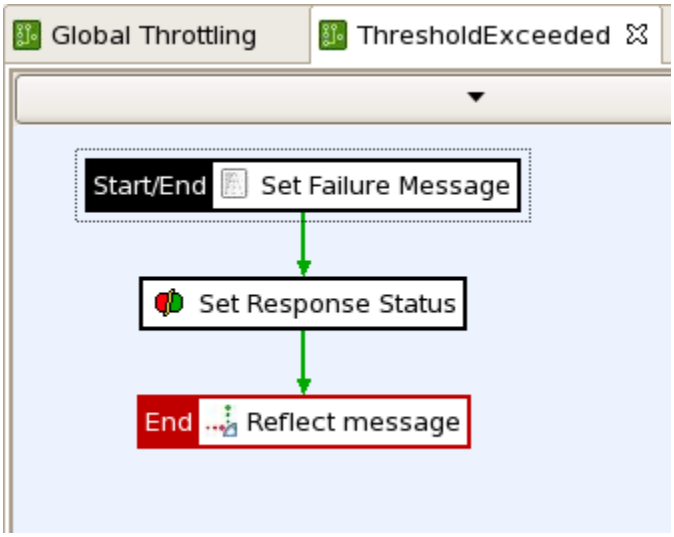
- This dialog box lets you control the number of messages processed in a time period. In this case, low values have been used to make it easy to demonstrate the effect without load.
- The “Track per key” check box is not selected, because you want all requests to be processed by this throttling filter and increment the rolling count irrespective of the service, user, operation, or protocol.
- The cache settings are used to keep the “request count.” The above case makes use of a default local maximum messages cache for convenience. This is local to the Gateway instance and is part of a default Gateway installation.

b. Click Cancel to close the editor.

Other filters in the policy include the following:

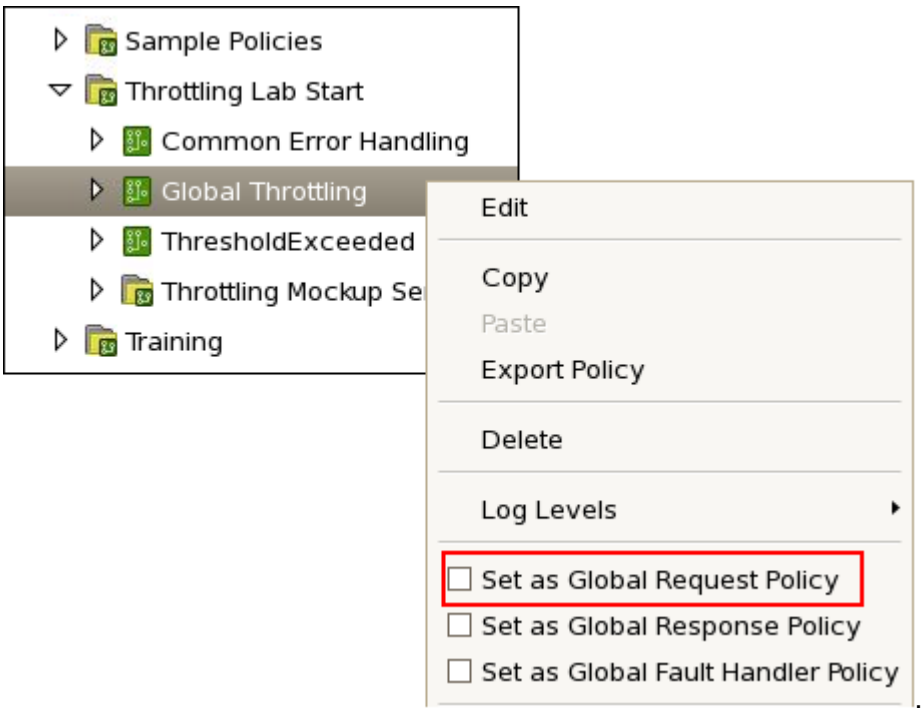
- The False filter is used to visually indicate that you will trigger the policy fault handler.
- The True filter is used to visually indicate that the execution continues if the limit has not been exceeded.
- The Call **Threshold Exceeded** filter, a Policy shortcut, makes a call to the Threshold Exceeded policy, which is a fault handler.

4. Select the Threshold Exceeded policy to view the circuit in the Policy Editor:



- The “Set Failure Message” filter returns a simple XML message.
- The “Set Response Status” filter ensures the correct reporting of the message blocking in real-time monitoring.
- The “Reflect” filter sets the HTTP return code to 403.

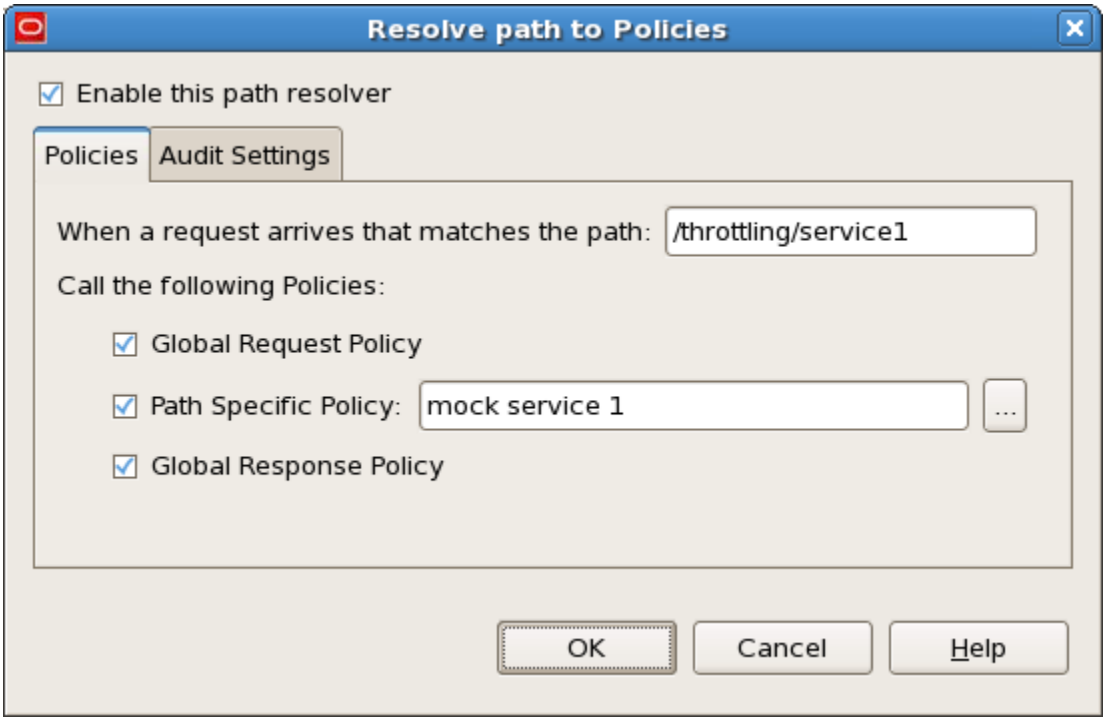
5. Right-click the Global Throttling policy and set it as the Global Request Policy. The policy icon will be updated to indicate this change.



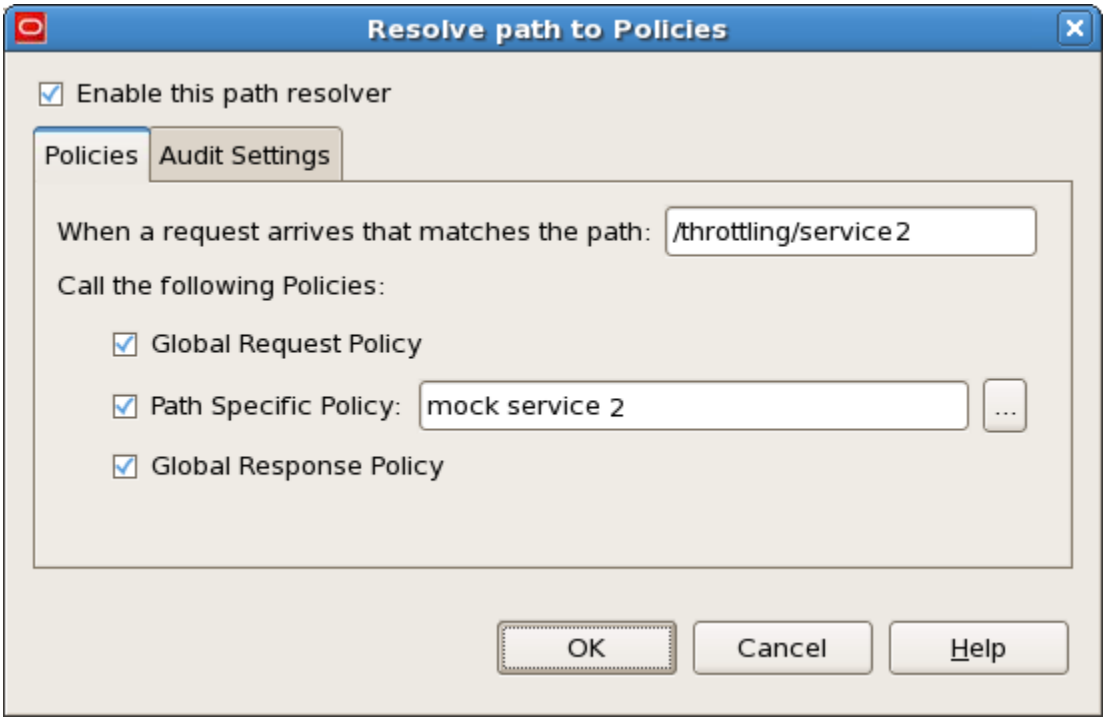
Mapping policy to the Relative Paths

Next, you will use two mock services to test the global throttling policy. The two mock services are imported with the throttling policies, under Policies > Throttling Lab Start > Throttling Mock Services. You can open and view their circuits.

6. Add a Relative Path under the Default Services node, name it `/throttling/service1` and configure it to call the mock service 1 policy, which is located in the container Throttling Lab Start > Throttling Mockup Services. Make sure the Global Request Policy option is selected.



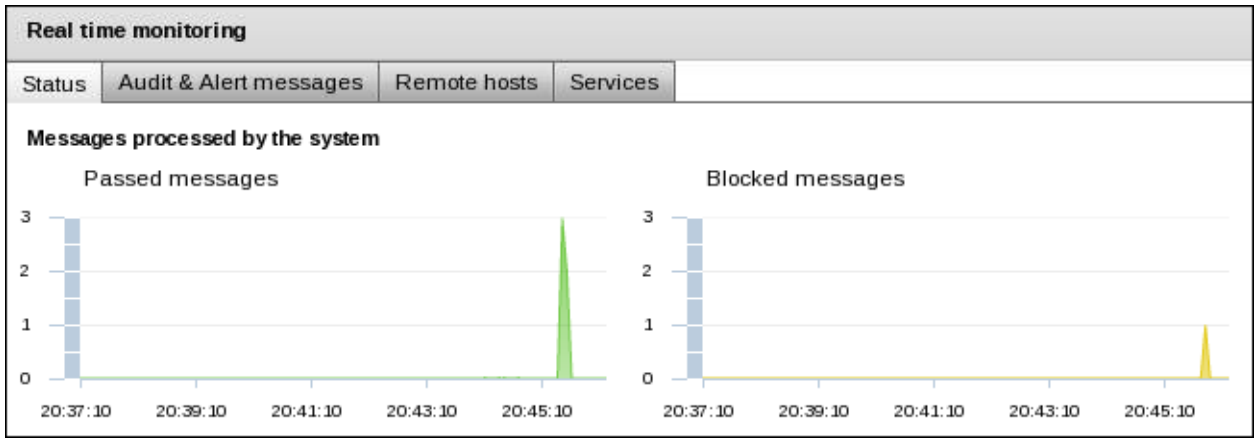
7. Similarly, add a Relative Path for mock service 2 with name `/throttling/service2`.



8. After configuring the two Relative Paths, deploy the configuration.

Testing and monitoring the global policy

1. In the browser, invoke the mockup service1 using the following URL: http://localhost:8080/throttling/service1
2. You can send multiple requests by clicking the reload button, and check how successful messages are returned 5 times, and failing messages (Throttling Limit Exceeded) are returned after that. Wait for a new 10-second period to start, and the service works again.
3. You can open Real time monitoring to track successful requests and failures.
   a. Go to the Gateway Management home page (http://localhost:8090) and click the Real time monitoring console link.
   b. You will see a graph similar to the image below:



4. In a similar fashion, you can run the test by using mock service2. The URL is: http://localhost:8080/throttling/service2

   Verify that the global limit is enforced across both services.

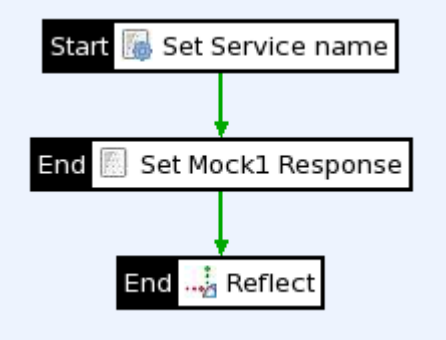
Practice 9-3: Applying Throttling at Service Level

Overview

Thus far, you have applied restrictions at a global level, something critical to protect systems from DoS attacks, for example. In this practice, you now apply a service-level policy to the mock1 service.

Tasks

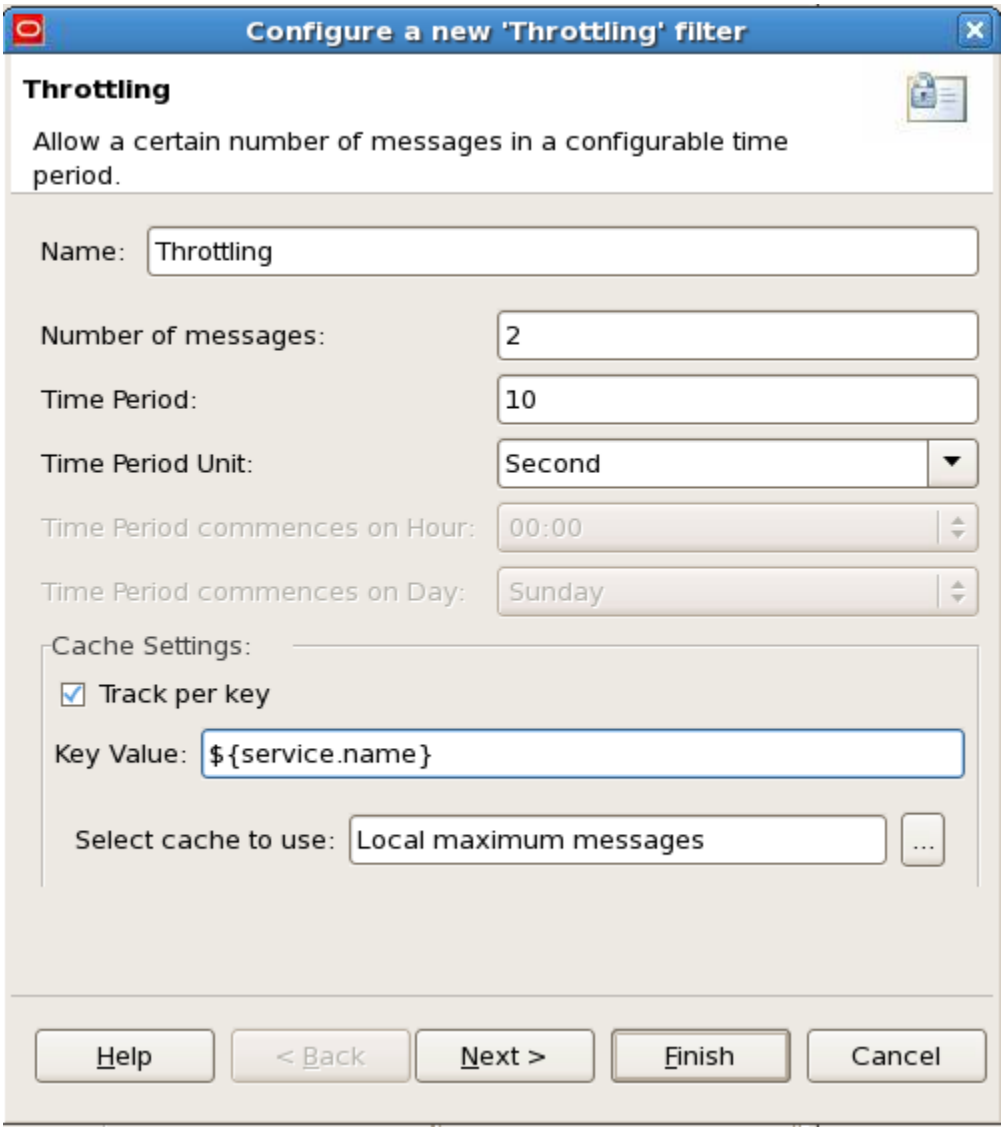
1. Open Policy Studio, expand Policies > Throttling Lab Start > Throttling Mock Services node.
2. Select Mock Service 1 policy. The circuit is displayed on the Policy Editor canvas as shown below:



3. In the Filters Palette, locate the Throttling filter (under Content Filtering), and drop it on the canvas.

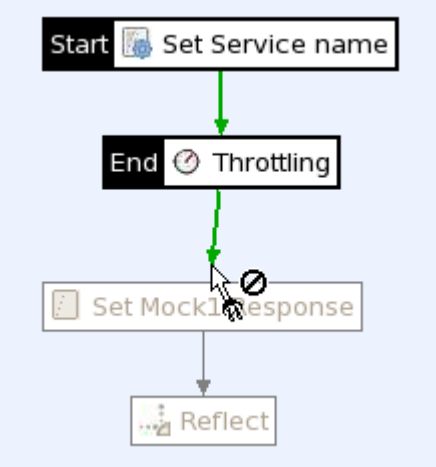
4. In the Configure a new ‘Throttling’ filter dialog box, make the following configuration:
   - Set the limits to 2 per 10 seconds.
   - Select the Track per key option and enter `${service.name}` as the tracking (unique) key in the Key Value field. The service name is set by the Set Service name filter and is a unique business name for a service.

   Note: Note that this name is totally independent from the request type or transport protocol.



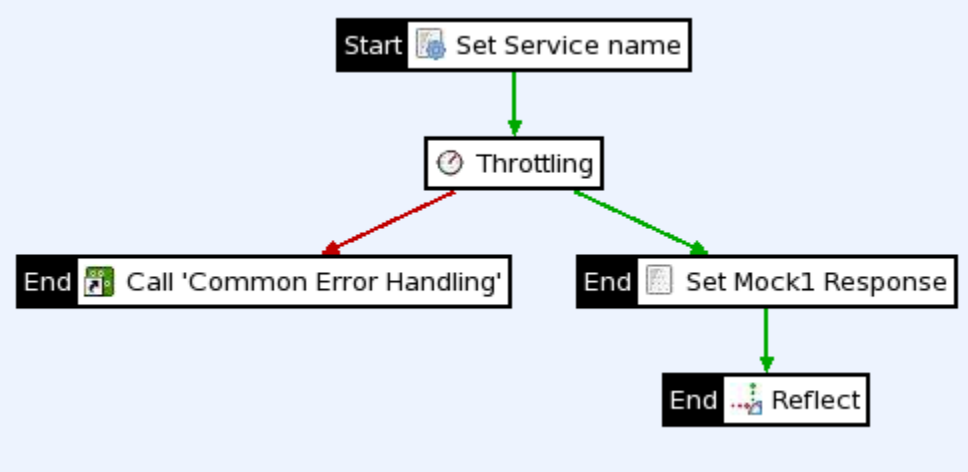
5. Click Finish. You should see the Throttling filter displayed in gray on the canvas.
6. The Throttling filter should be added between the Set Service name and Set Mock1 Response filters. Perform the following steps:
   a. Select the connector from Set Service name to Set Mock1 Response, and link it to the Throttling filter. Now the Set Mock1 Response and Reflect filters are in gray.

b. Select Success Path from the Filters Palette, then select the Throttling filter, and link to the Set Mock1 Response filter as shown below:



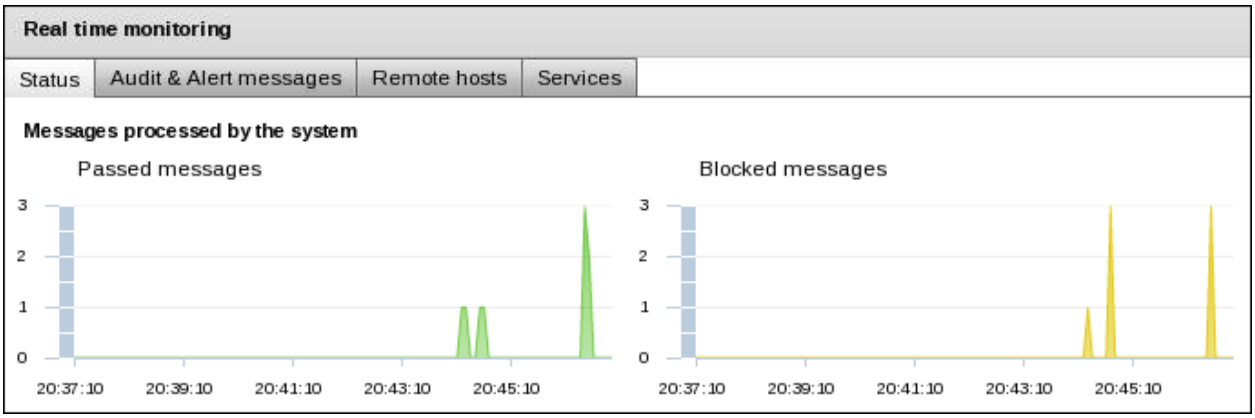
7. Drop a Policy Shortcut filter onto the canvas.
8. In the Configure a new 'Policy Shortcut' filter dialog box, select the Common Error Handling policy provided to you.
9. Wire this filter to the failure path of the Throttling filter. The common error handling policy uses the service name to return an appropriate error message and set the monitoring status to fail.

   The final policy will look like this:



10. Deploy the configuration.
11. Go to the browser, and test Service1 again. After two requests, Service1 will be blocked but Service 2 will continue to work until the global policy limit (5) has been reached.

12. Again, use real-time monitoring to visualize the behavior.



Practices for Lesson 10: Configuring SSL

Chapter 10

Practices for Lesson 10: Overview

Practices Overview

In these practices, you will set up a new SSL listener on the OEG gateway. You will create the necessary certificate/key pairs for this lab and then set up mutual SSL. The tasks you will perform include:

- Create certificates (self-signed CA certificate and server certificate)
- Create an HTTPS interface
- Export and import certificates and keys
- Setup Mutual SSL

Practice 10-1: Creating the Certificate Authority (CA)

Overview

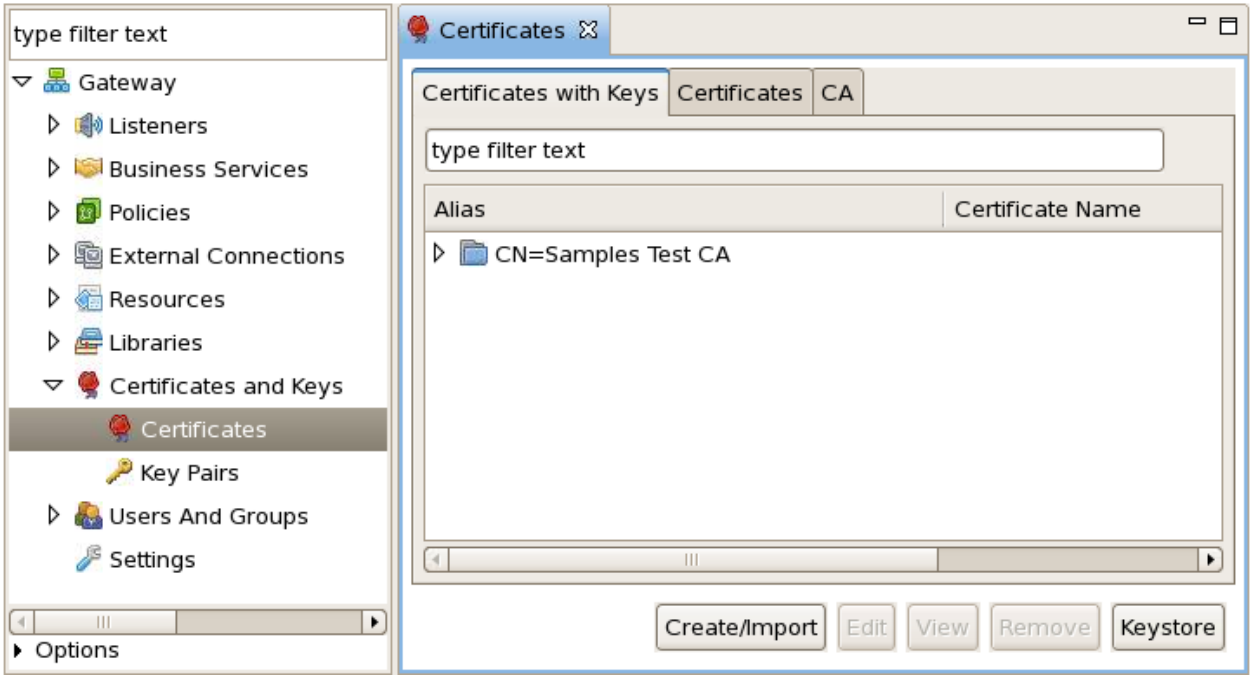
In this practice you create a self-signed CA certificate, which will be used to sign other certificates later.

Assumptions

The Enterprise Gateway is up and running.

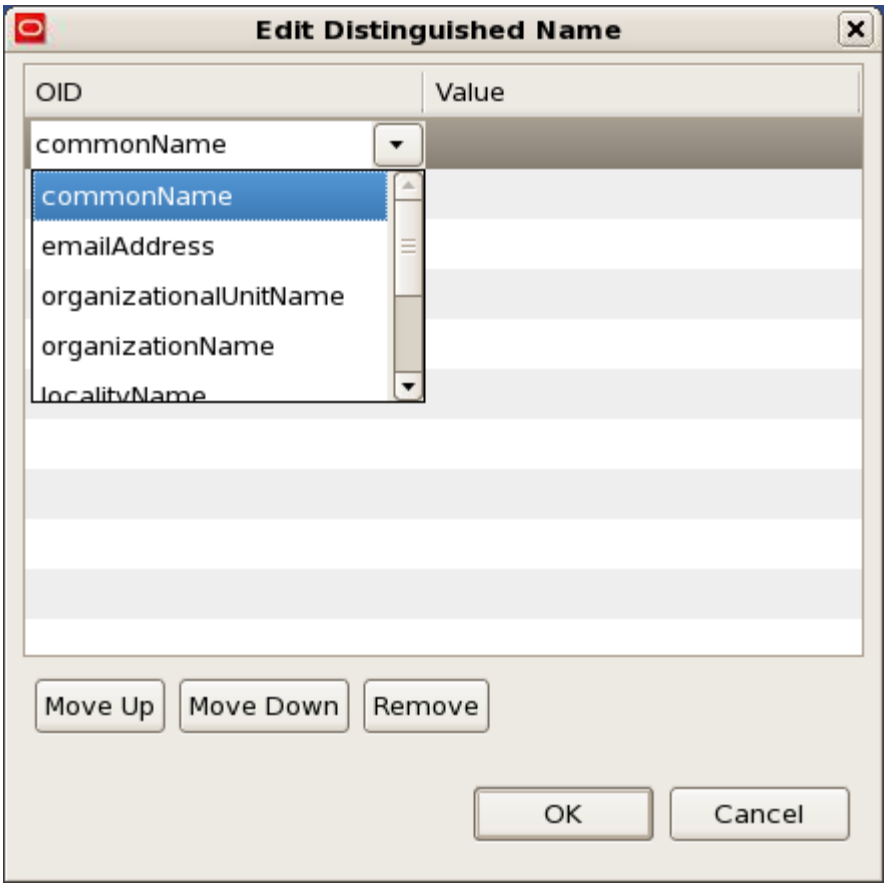
Tasks

1. In Policy Studio, expand the Certificates and Keys node in the left frame, and select Certificates.



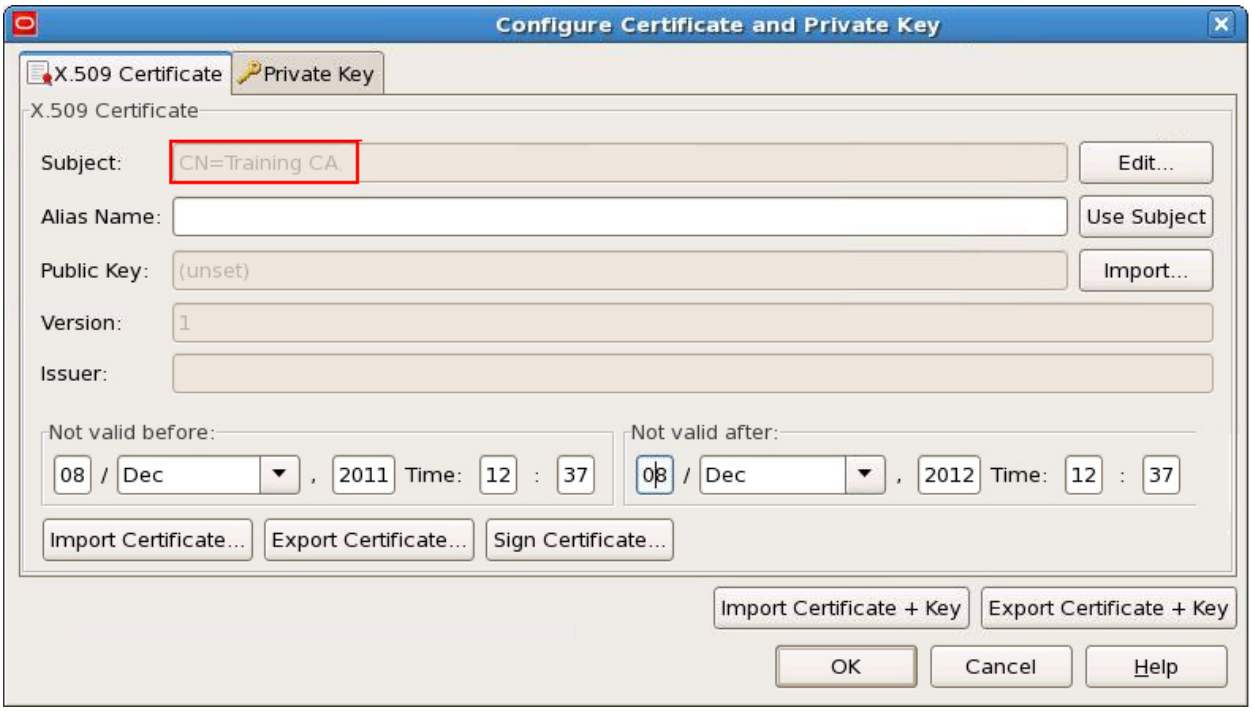
2. Click Create/Import. The Configure Certificate and Private Key window is displayed.

3. Click Edit to the right of the Subject field. The Edit Distinguished Name dialog is displayed. Specify the certificate basic information as follows:
   a. Select commonName for OID.



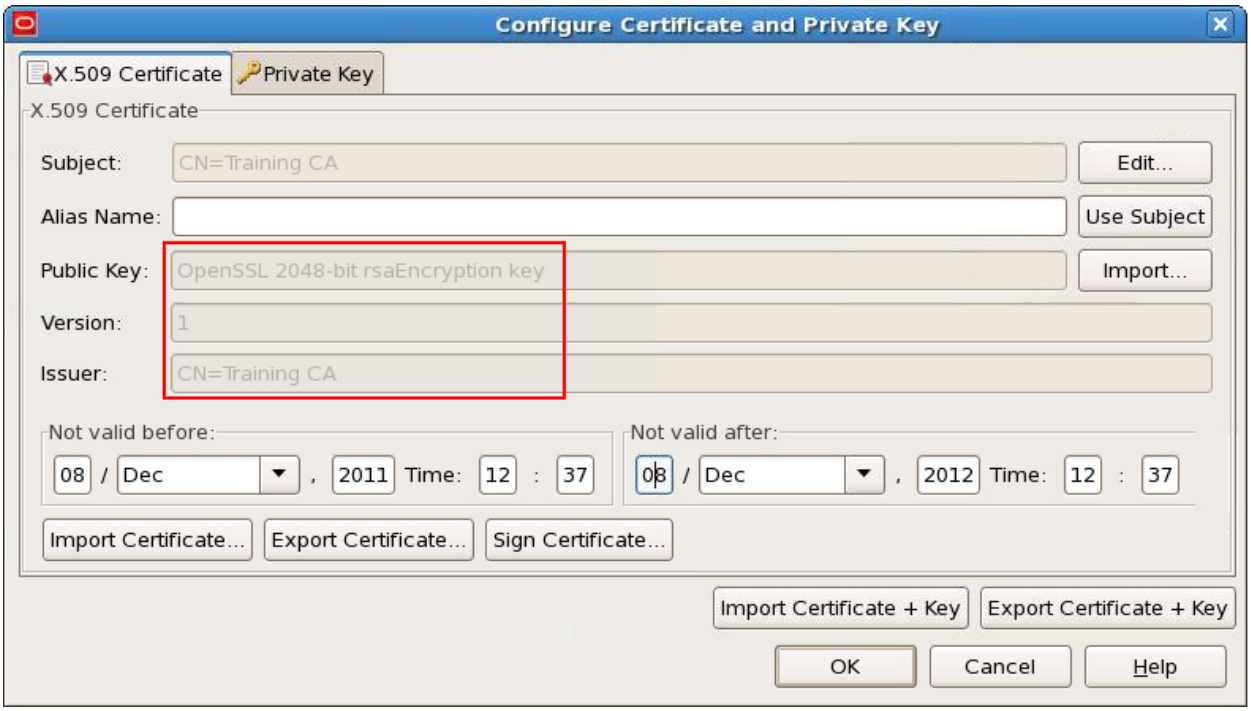
   b. Enter `Training CA` as the value.

c. Click OK. The Subject field is updated with the values you entered.

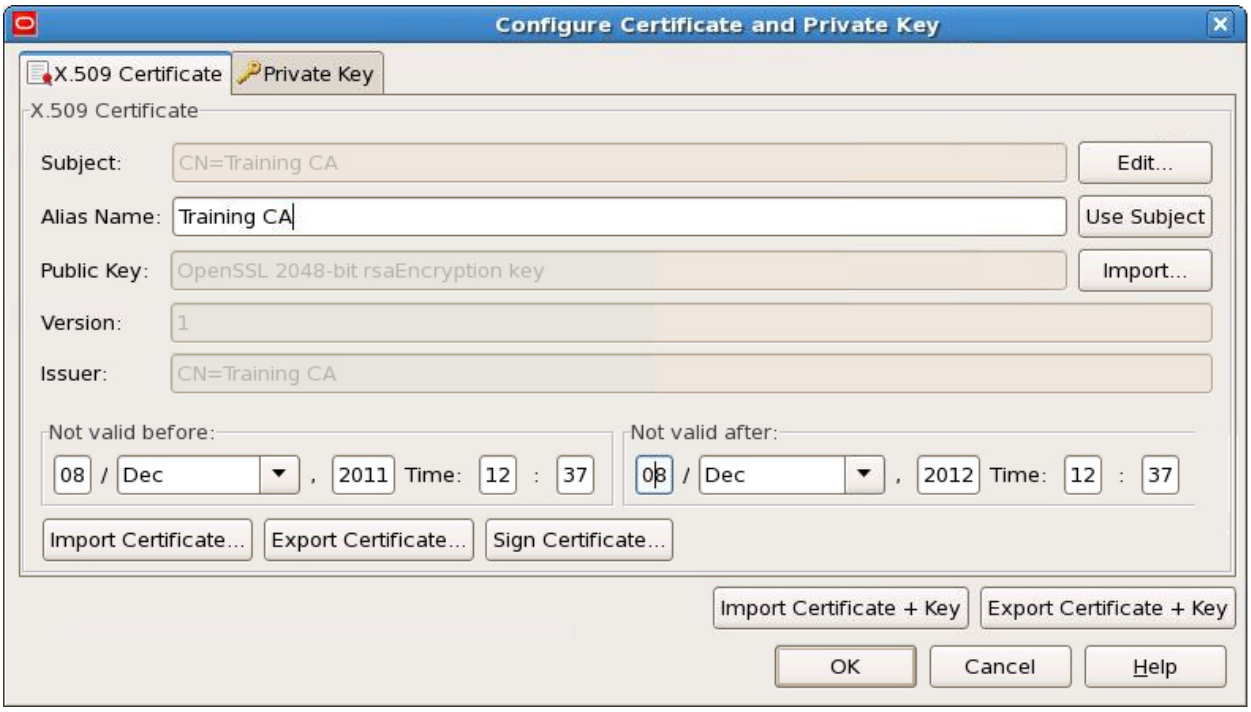


4. Click **Sign Certificate** to self-sign the certificate.
   a. Select Yes when you are asked “Do you want to self-sign the certificate?”
   b. Select Yes when you are asked “Do you wish to generate a key-pair?” in the Public/Private Key Missing dialog box.

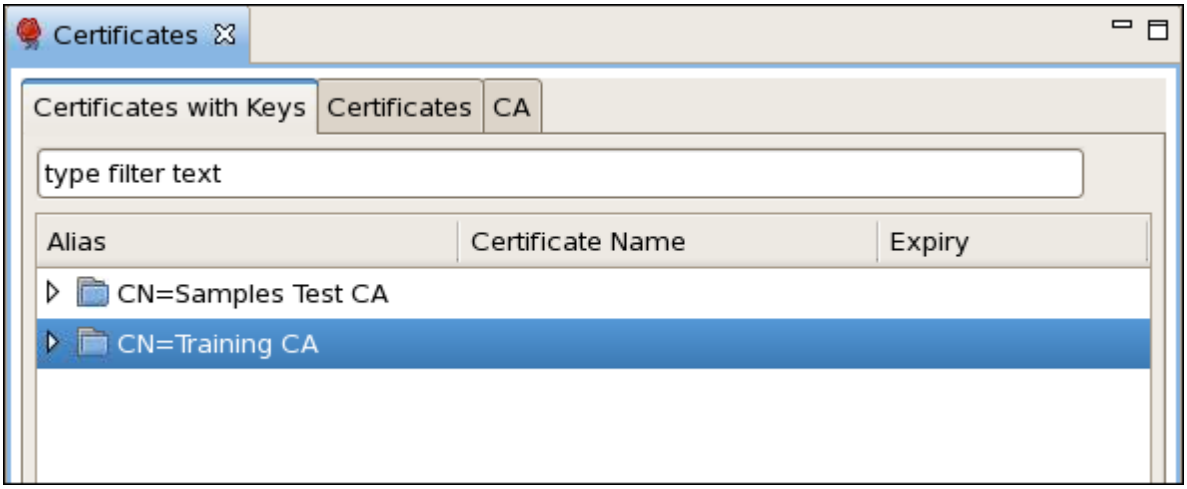
   Note that the Public Key and Issuer fields are updated.



5. Finally, enter “Training CA” as the alias name, and click OK.



6. You now have created the Training CA certificate with associated private keys in the Certificates store.



Practice 10-2: Creating the OEG Server Certificate

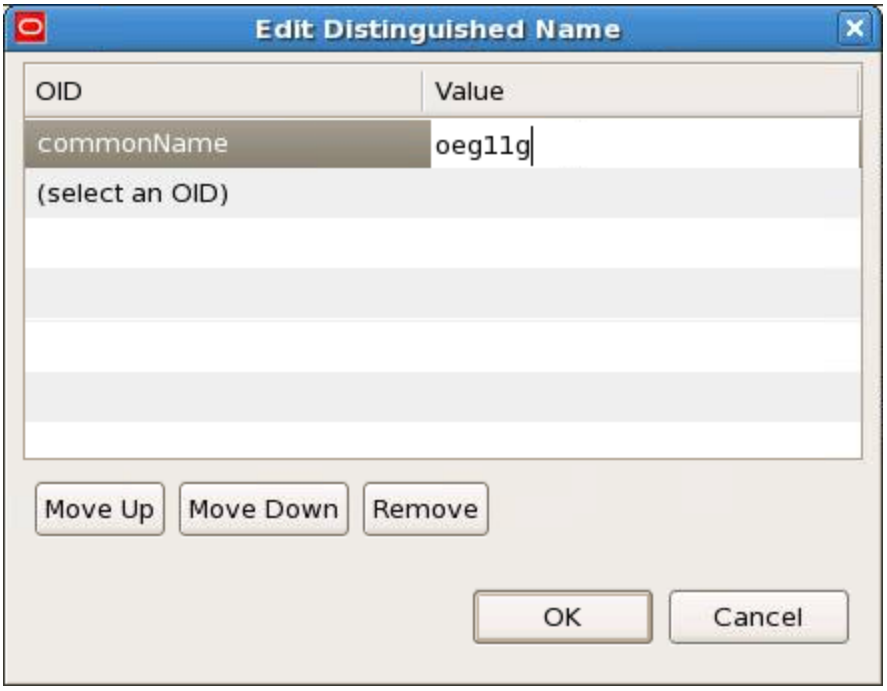
Overview

In this practice you use the CA you have just created to sign a certificate for the Enterprise Gateway. The server name you will use to invoke the service is `oeg11g`, so the server certificate commonName attribute should match this value.

The way you create the server certificate is very similar to what you did for the CA certificate.

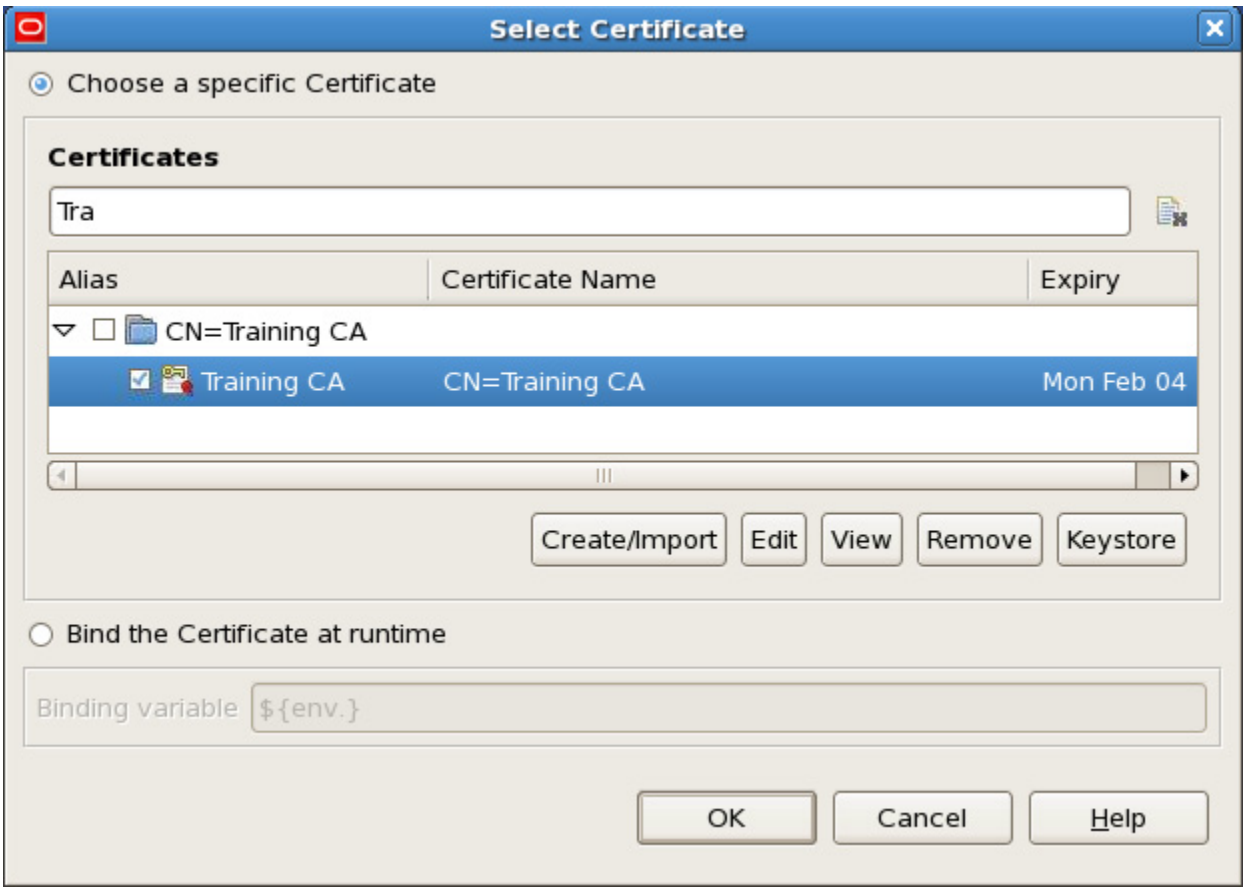
Tasks

1. In the Certificates main panel of the Policy Studio, click **Create/Import**. The Configure Certificate and Private Key window is displayed.
2. Click **Edit** to the right of the Subject field. Set commonName to **`oeg11g`** in the Edit Distinguished Name window.



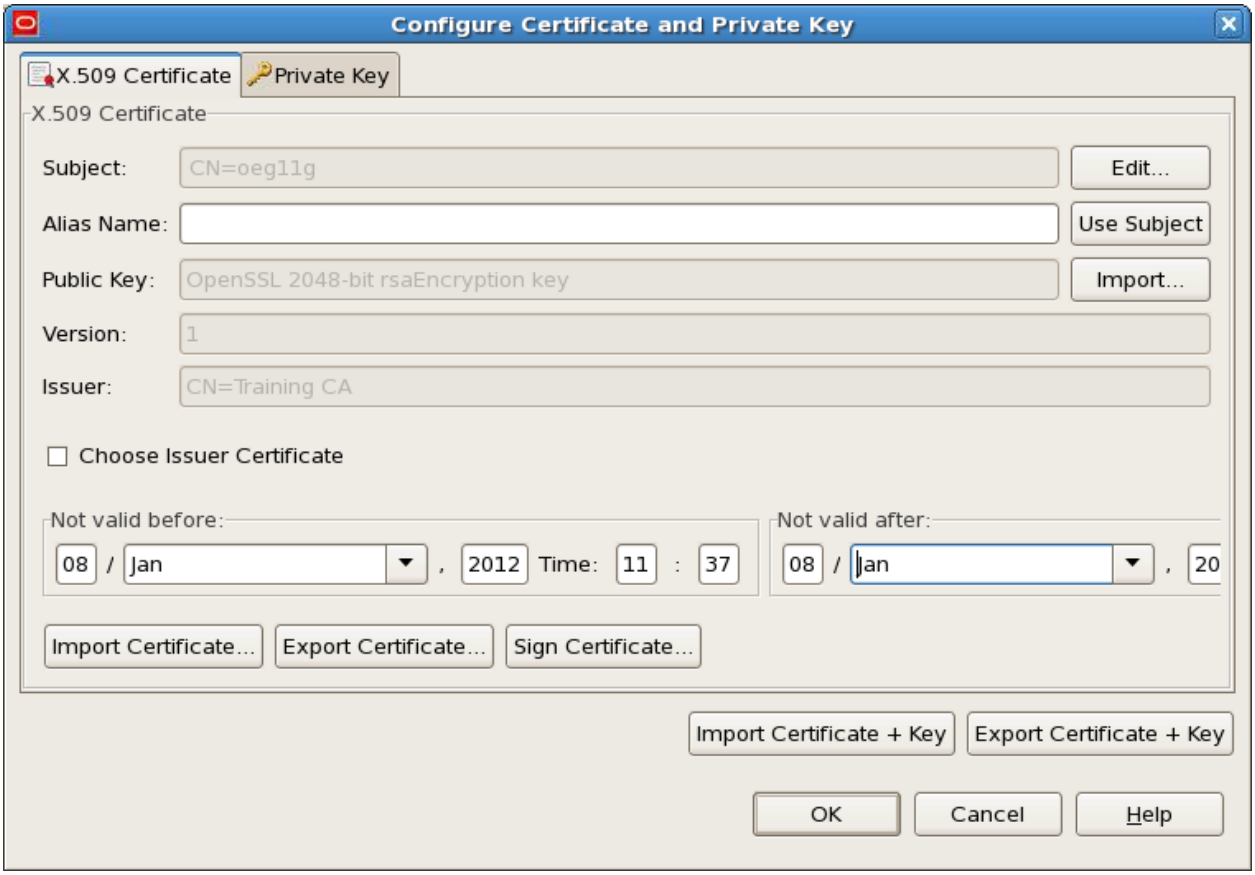
3. Click **OK**. The Subject field is updated with the values you entered.
4. Click **Sign Certificate** to sign this server certificate with the CA certificate you created previously.
   a. Select **No** when you are asked “Do you want to self-sign the certificate?” in the Self-sign Certificate dialog box.

b. Select the Training CA certificate from the certificates list by checking the box next to the certificate.



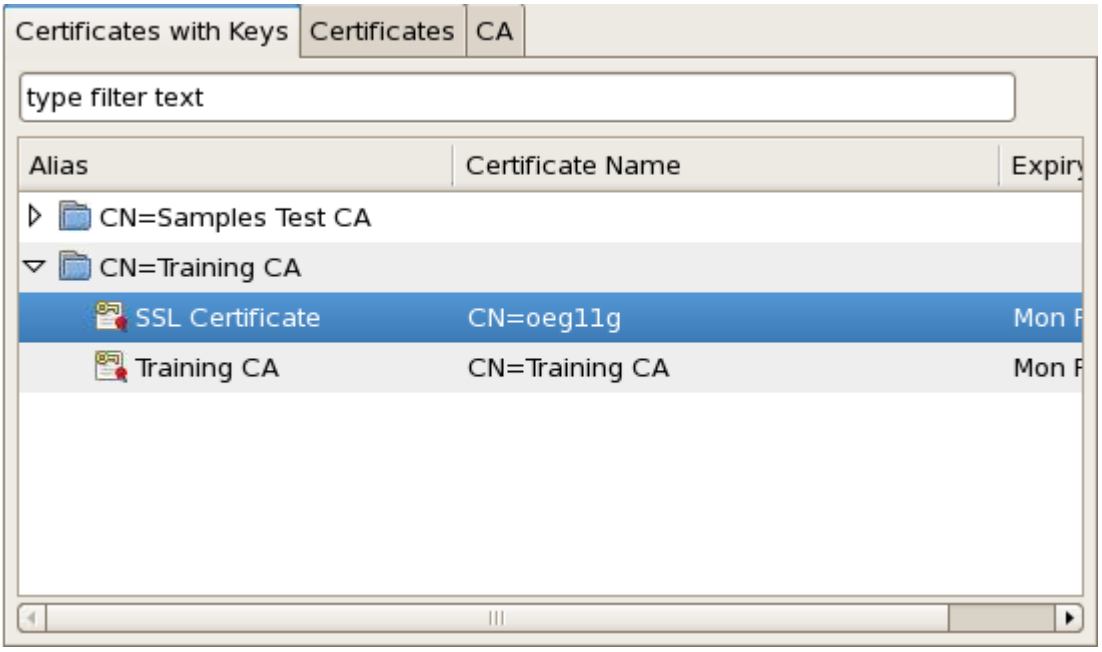
c. Click OK.

d. Select Yes when you are asked “Do you wish to generate a key-pair?” in the Public/Private Key Missing dialog box. Note that the Public Key and Issuer fields are updated.



e. Enter “SSL Certificate” as the alias name, and click OK.

f. Now you have two certificates ready to use – the Training CA and the SSL Certificate.



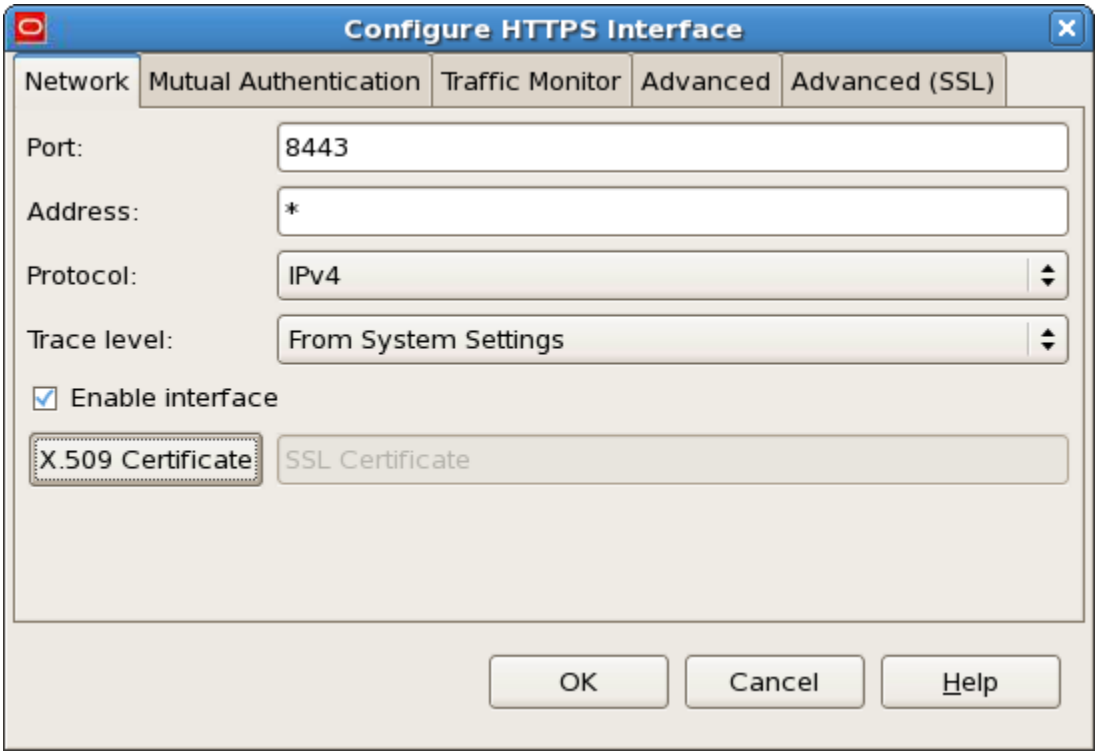
Practice 10-3: Creating an HTTPs Listener

Overview

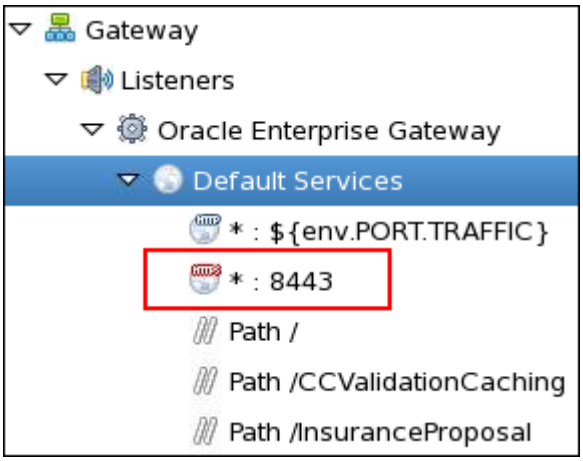
By default, the Enterprise Gateway is configured to listen for requests on a single port (generally 8080) in HTTP mode. In this practice you will add a new listener interface that takes HTTPs requests, and then test the SSL connection by using a browser.

Tasks

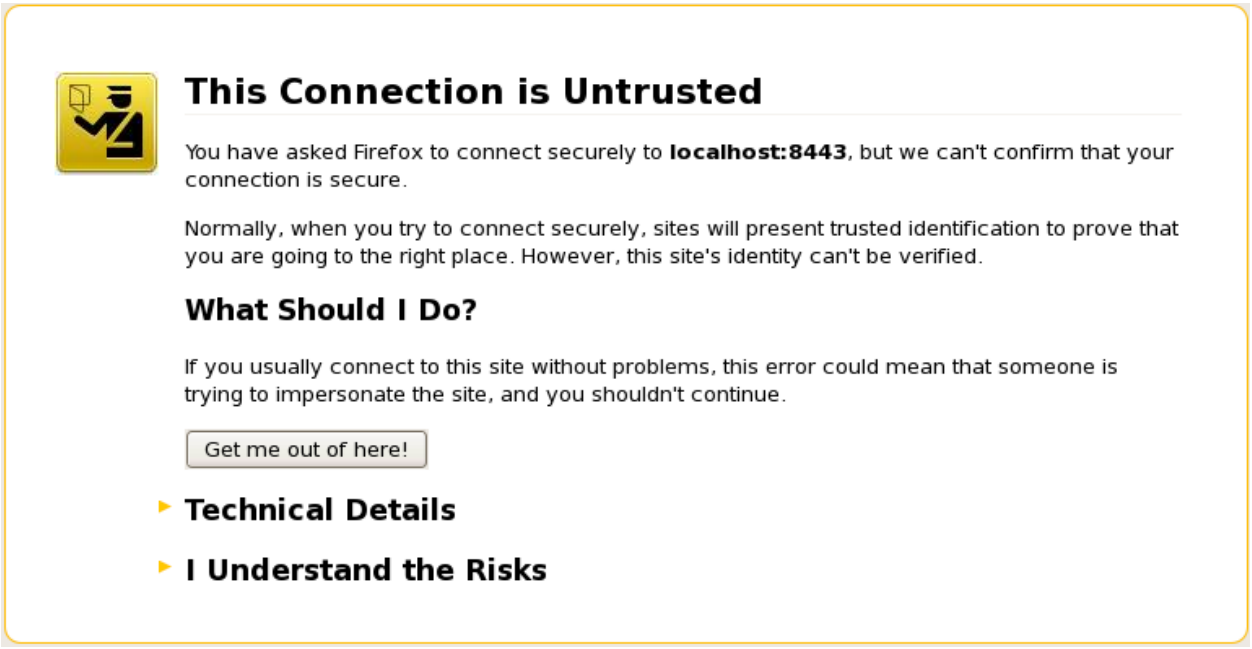
1. In Policy Studio, navigate to Listeners > Oracle Enterprise Gateway > Default Services, right-click and select **Add Interface** >**HTTPS** from the drop-down menu. The Configure HTTPS Interface window is displayed.
2. Configure the port as 8443, and use the SSL server certificate – `SSL Certificate` you just created as the X509 certificate. You can leave all other values unchanged.



3. Click OK. You should see the new interface added under the Default Services node.



4. Deploy the server configuration.
5. To test the SSL connection from Firefox, perform the following steps:
   a. Open Firefox browser and enter the following URL in the address field: `https://localhost:8443/healthcheck`. You will see security warnings resembling the following screen shot because the Training CA for this server was self-signed.



   b. Click the “Technical Details” link to get more information.
   c. Click the “I Understand the Risks” link, and then the Add Exception button.

d. In the Add Security Exception window, click Get Certificate, and then view the Certificate Status.

e. Click Confirm Security Exception to accept this as an exception. You should see the healthcheck status change to OK.

Practice 10-4: Setting Up Mutual SSL (optional)

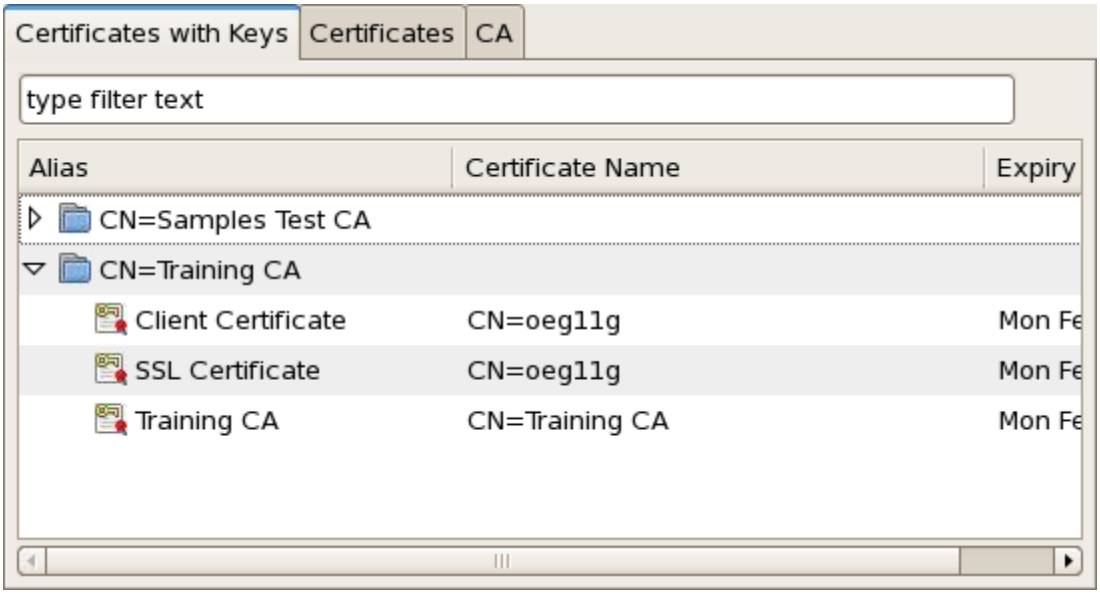
Overview

In this practice you will set up a mutual SSL.

Note: To make the practice a bit more challenging, the instructions are provided at a high level, instead of step by step.

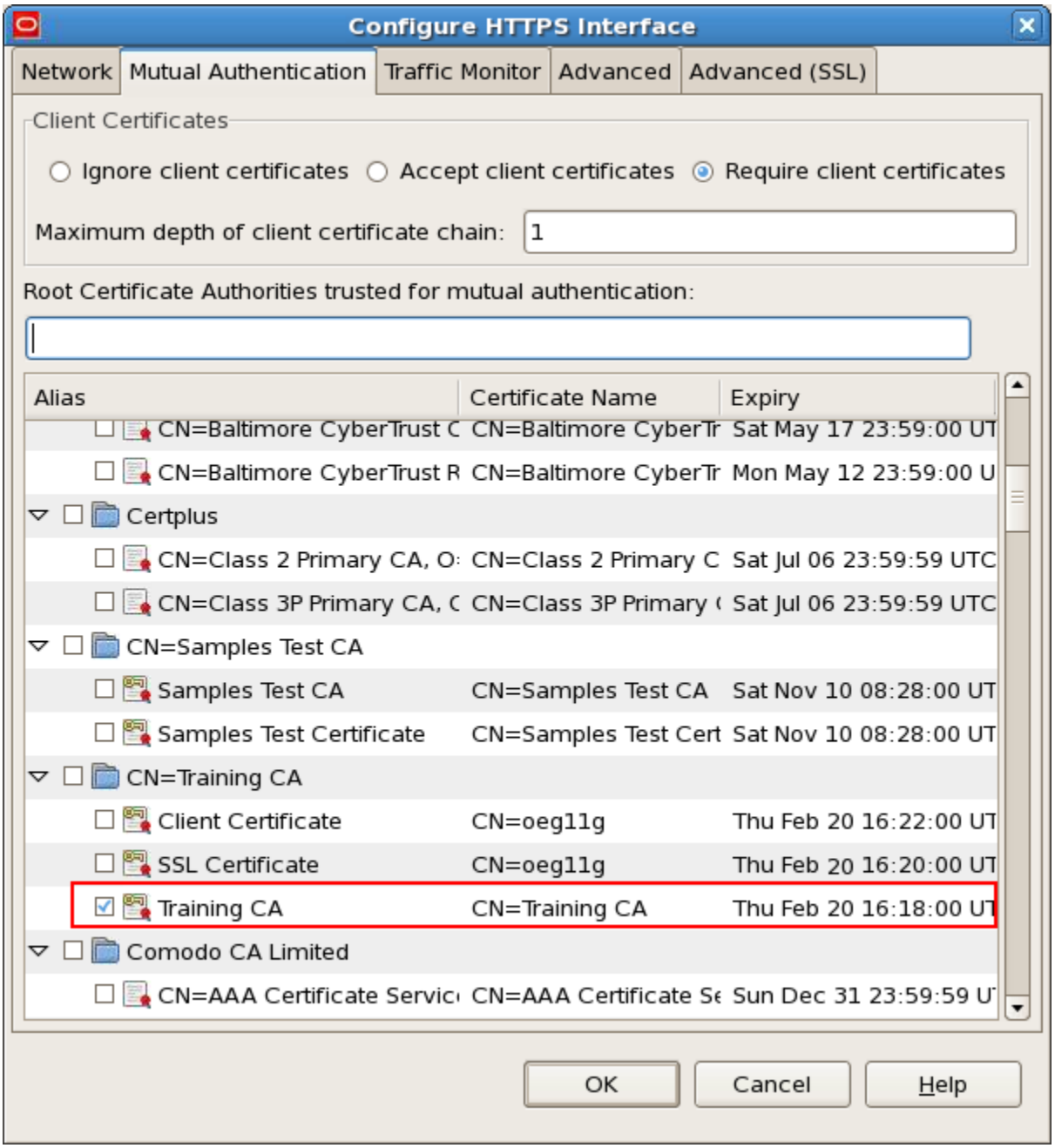
Tasks

1. Create a certificate and private keys for yourself by using the same steps you followed to create the SSL server certificate. Make sure to sign the certificate with the Training CA certificate you created in Practice 1. Below is an example of what you will see in the Certificate store after creating the client certificate:



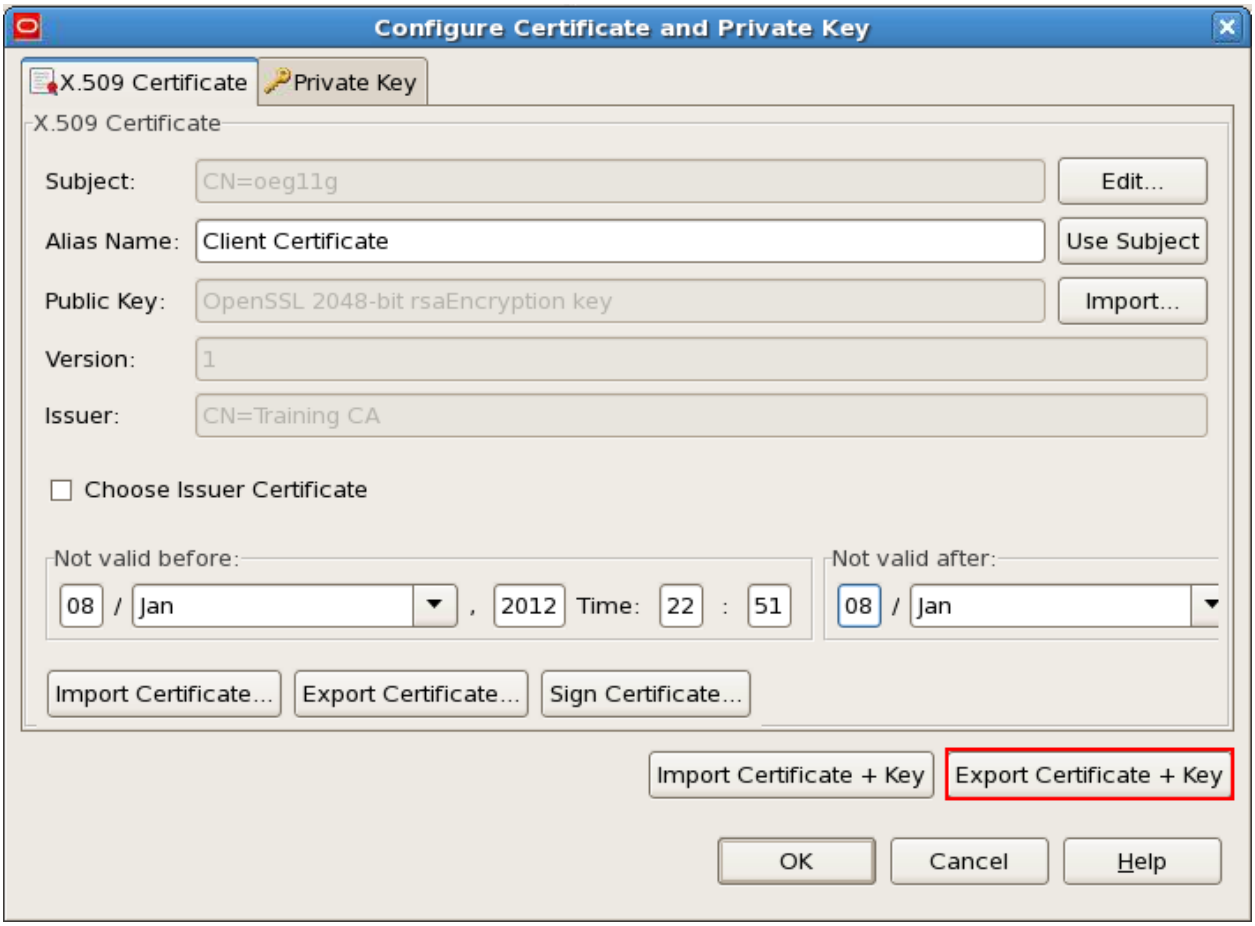
2. Edit the HTTPs Interface you created previously.
   a. On the Mutual Authentication tab, select the Require client certificates option.

b. In the list of trusted certificates, choose Training CA as the Root Certificate.



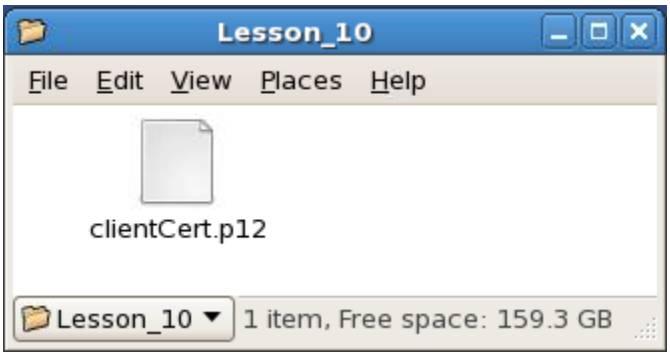
3. Deploy the configuration to the OEG Gateway.

4. Export your client certificate and private keys from Policy Studio in **P12** format to the /home/oracle/labs/Lesson_10 folder (make sure to select Export Certificate + Key)



Note: You need to provide a password to encrypt the certificate and key when exporting them. Make a note of the password, and you will need it when importing the certificate and key to the client.

5. If exporting is successful, you should see the certificate and key in the /home/oracle/labs/Lesson_10 folder.



Testing

To test whether mutual SSL works correctly, you can use Firefox.

1. Import personal certificate and private keys into Firefox.

a. Open Edit menu, navigate to Preferences > Advanced > Encryption), and select View Certificate.

b. In the Certificate Manager window, import the certificate you just exported.

2. Now invoke the healthcheck service again, and you will be prompted for the client certificate.

Practices for Lesson 11: Securing XML Messages

Chapter 11

Practices for Lesson 11: Overview

Practices Overview

In these practices, you will learn to use XML signature and encryption policy to protect the integrity and confidentiality of your XML messages.

Practice 11-1: Creating Client Certificate and Key

Overview

In this practice you create the client certificate and key, which will be used when signing a client request.

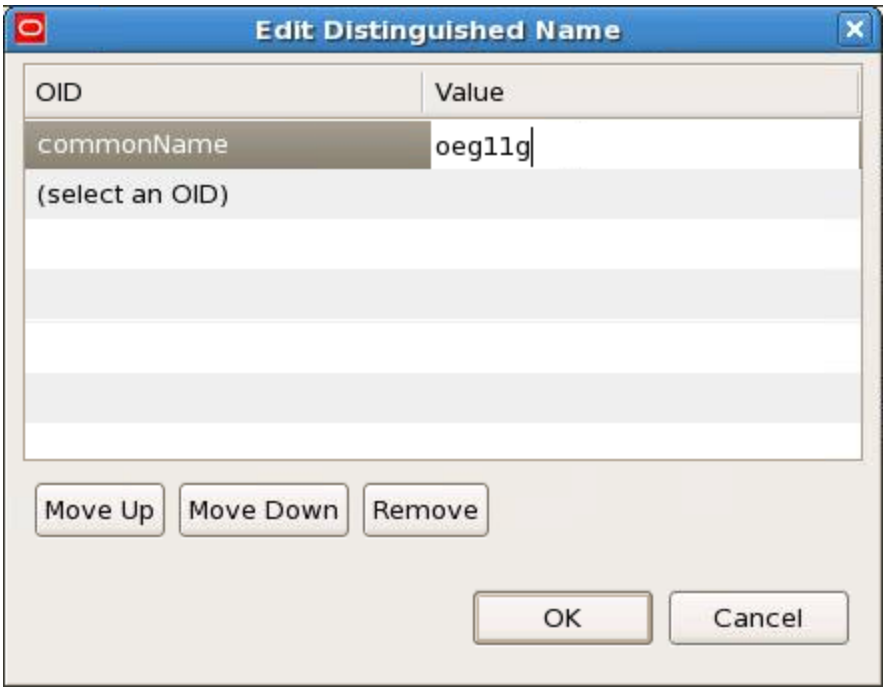
Note: If you complete Practice 10-4: Setting Up Mutual SSL, successfully exporting a client certificate and key, you can skip this practice to proceed to Practice 11-2.

Assumptions

The Enterprise Gateway is up and running.

Tasks

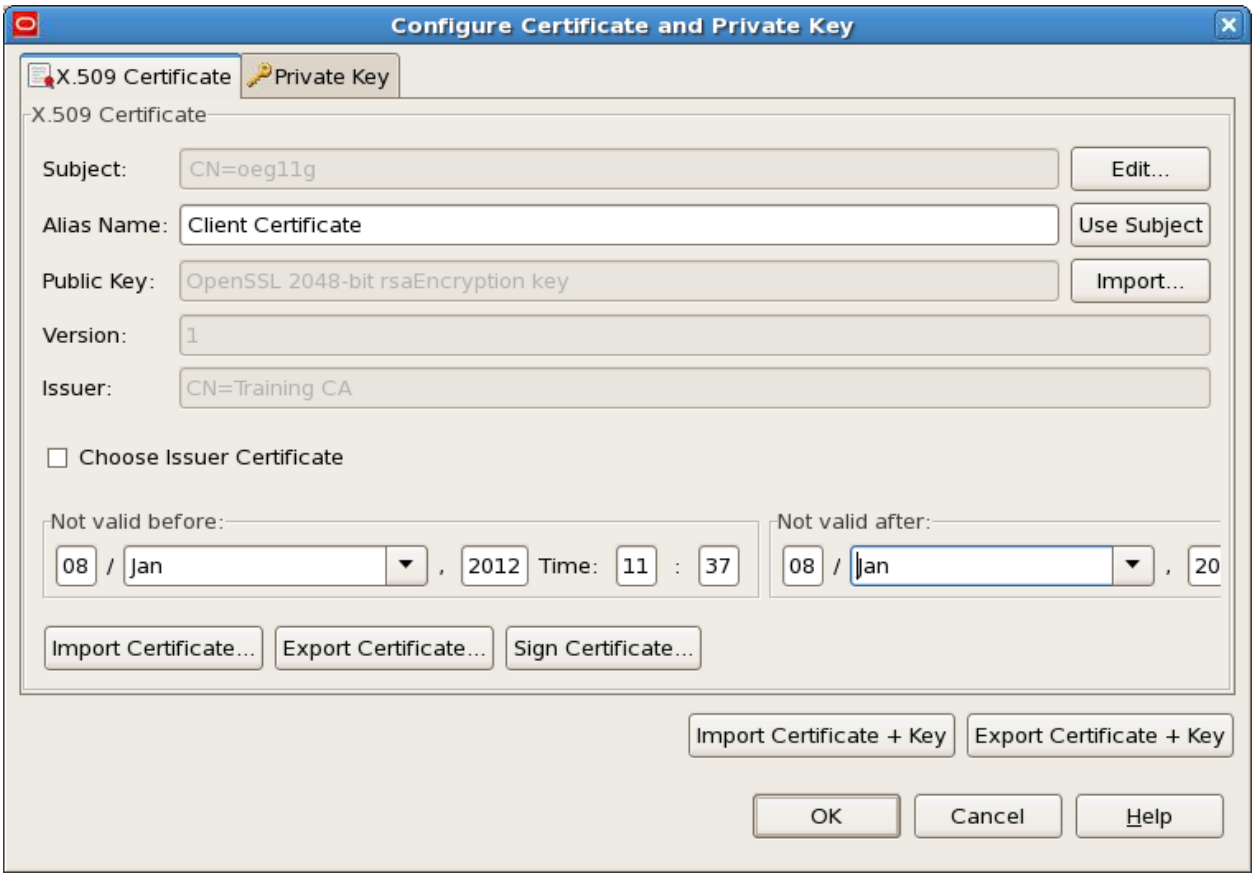
1. In Policy Studio, expand the Certificates and Keys node in the left frame, and select Certificates.
2. In the Certificates main panel, click **Create/Import**. The Configure Certificate and Private Key window is displayed.
3. Click **Edit** to the right of the Subject field. Set commonName to `oeg11g` in the Edit Distinguished Name window.



4. Click **OK**. The Subject field is updated with the values you entered.
5. Click **Sign Certificate** to sign this server certificate with the CA certificate you created previously.
   a. Select **No** when you are asked “Do you want to self-sign the certificate?” in the Self-sign Certificate dialog box.
   b. Select the Training CA certificate from the certificates list by selecting the check box next to the certificate.
   c. Click OK.

d. Select Yes when you are asked “Do you wish to generate a key-pair?” in the Public/Private Key Missing dialog box. Note that the Public Key and Issuer fields are updated.

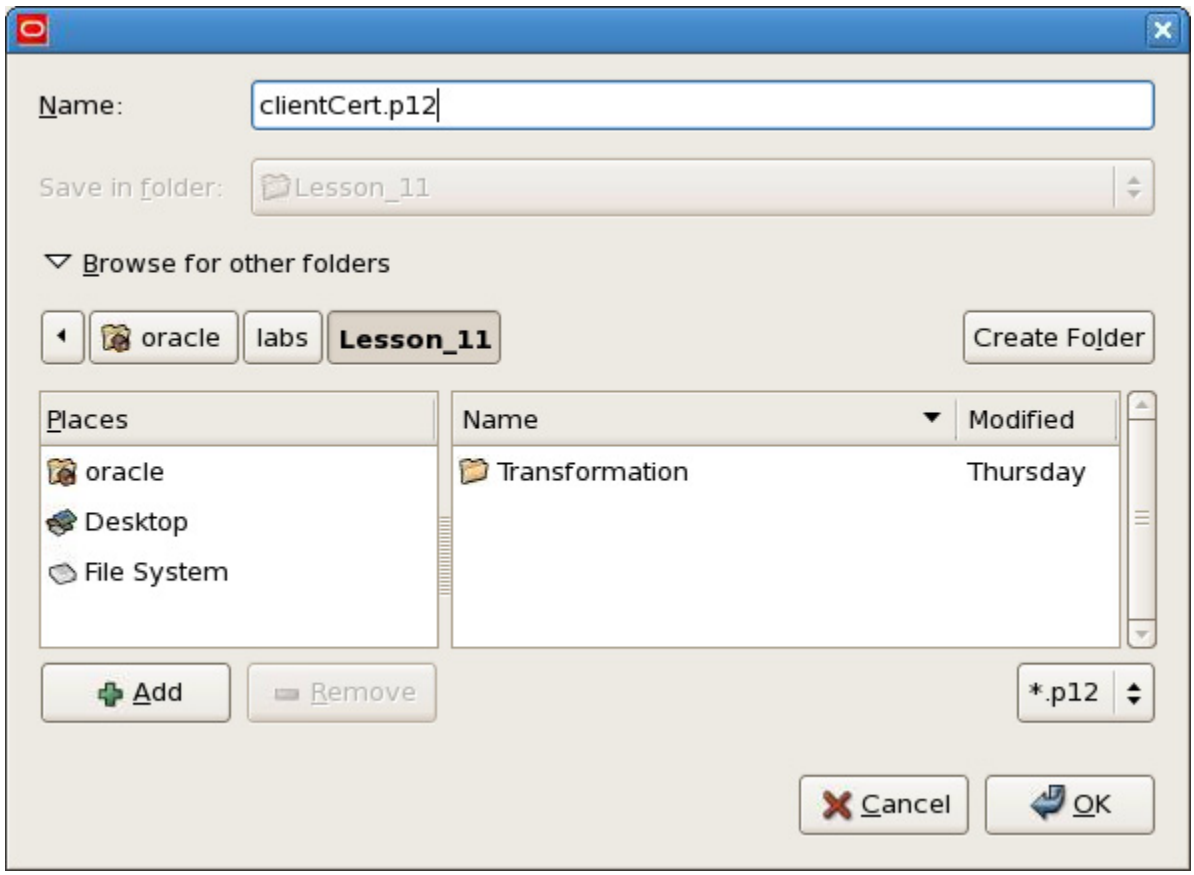
e. Enter “Client Certificate” as the alias name.



6. Export your client certificate and private keys by performing the following steps:

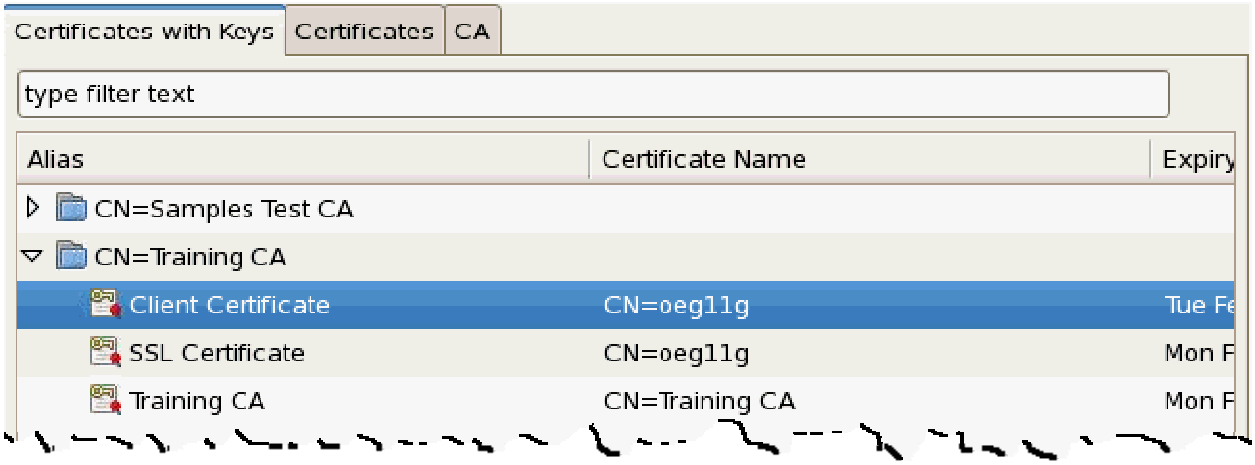
a. Click Export Certificate + Key. Make sure you have exported the private key as you will need it to sign a message in the next lab.

b. Navigate to the /home/oracle/labs/Lesson_11 folder, make sure the file format is P12, and enter the name `clientCert.p12`.



c. Provide a password to encrypt the certificate and key when exporting them. Make a note of the password; you will need it when importing the certificate and key to the client.

7. Click OK in the Configure Certificate and Private Key window after exporting the client Certificate and Key. Now you should see three certificates in the Certificates store – Training CA, SSL Certificate, and Client Certificate.



Practice 11-2: Securing XML Messages

Overview

In this practice you work with XML Encryption and Digital Signatures filters. You implement a common pattern, which includes receiving a certificate from a partner via a digital signature in the request and using it to encrypt the response.

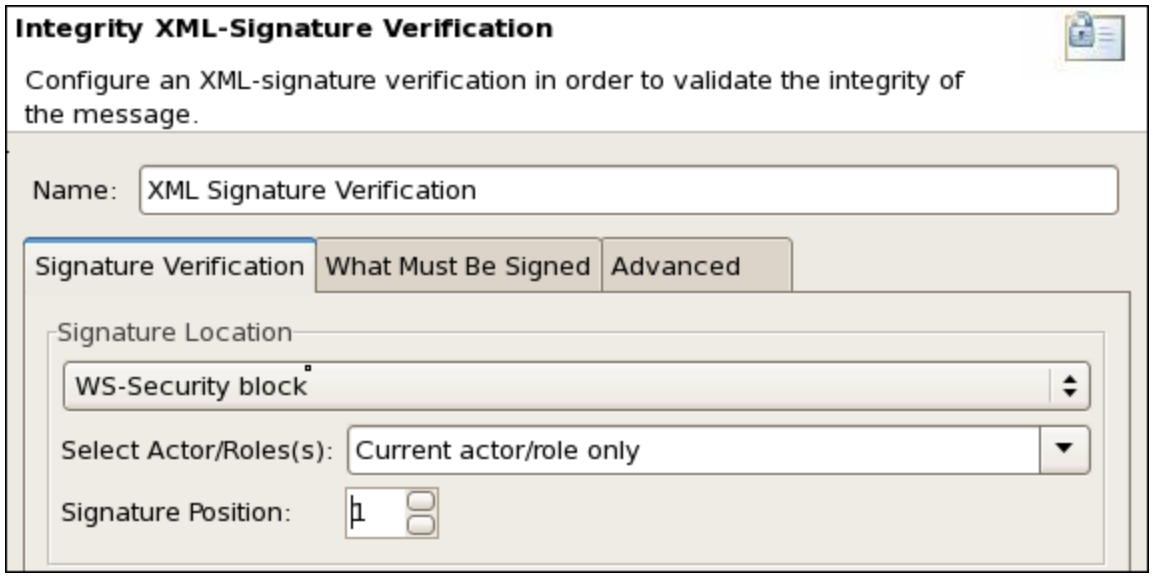
Verifying XML Signature

Tasks

Creating a policy

In this section, you create a policy to verify a digital signature.

1. In Policy Studio, create a policy and name it VerifySignature-Encrypt.
2. Drop an "XML Signature Verification" filter on the Policy editor canvas. This filter is located in the Integrity category.
3. You need to configure three settings in this filter:
   a. Signature Location - Use the signature in the first WSSecurity block for the current actor. You need to configure the filter as shown below:



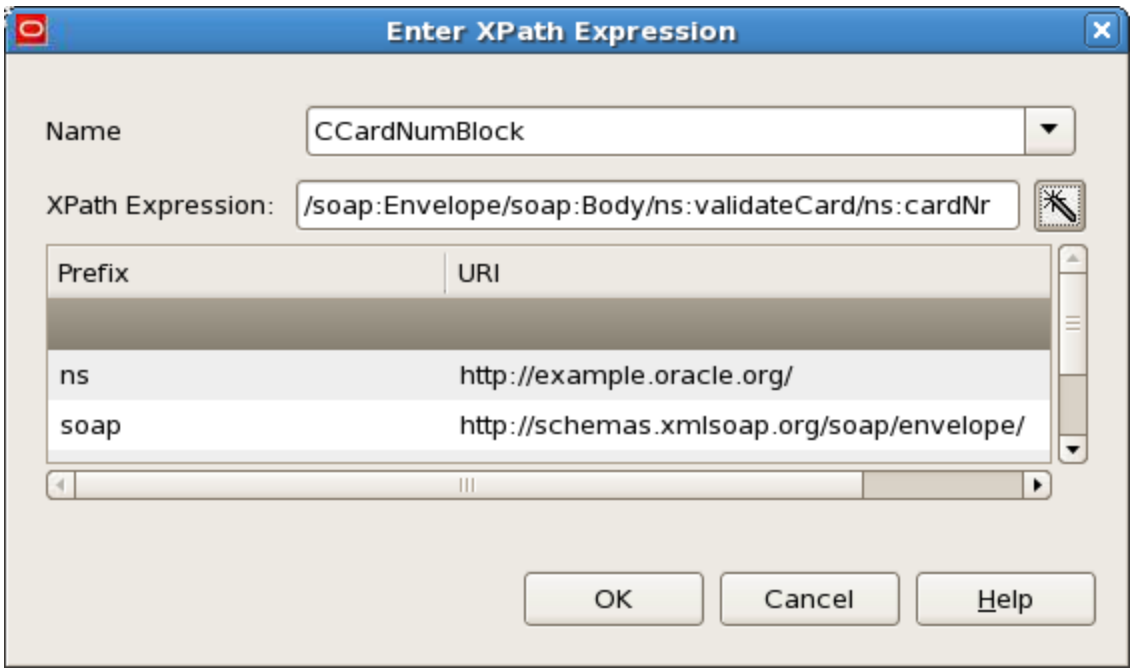
   b. Find Signing Key – Specify the location of the gateway where you will find the public key to verify the signature. For this practice, accept the default setting: Via KeyInfo in message.
   c. What must be signed? Any part of the message can be specified. You can either use the predefined Node locations, or specify a custom location via XPath. For this lab, you will specify a location that is the credit card number block. You, therefore, need to do the following:
      1) Click the "What Must Be Signed" tab and select the XPaths tab.
      2) Click Add.
      3) In the Enter XPath Expression dialog box, enter `CCardNumBlock` as the XPath expression name.

4) Click the wizard icon  .
5) In the XPath Wizard:
   a) Load the `ValidateCCRequest.xml` file from /home/oracle/labs/Lesson_11 (step 1 in the screenshot)
   b) Select the <ns:cardNr> element in the main window (step 2 in the screenshot)
   c) Click Use this path (step 3 in the screenshot)

d) Click Evaluate to check that the XPath expression returns what you need (step 4 in the screenshot)

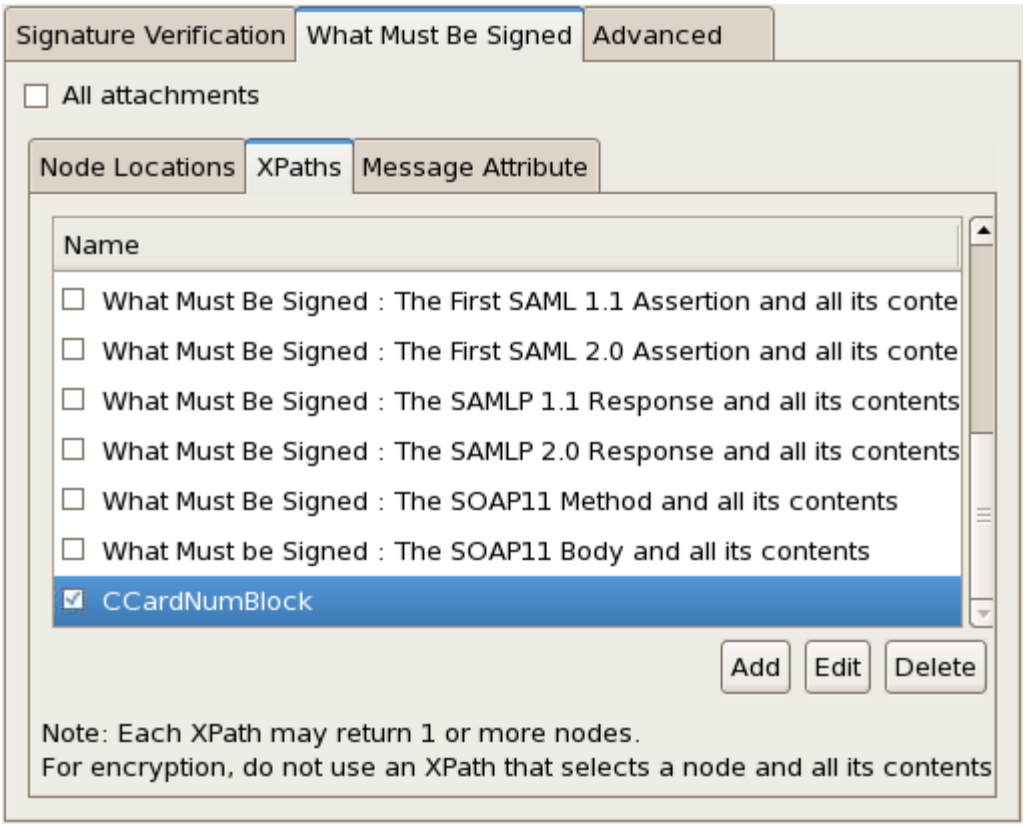
e) Click OK.

6) The Enter XPath Expression dialog box is updated with a newly defined XPath expression as shown below.



7) Click OK.

8) Select the CCardNumBlock XPath expression from the list.



a) Click Finish to close the filter editor.

4. Right-click the VerifySignature-Encrypt filter and set it as the start of the circuit.

5. Add a Reflect Message filter with a response code of 200 on successful verification of the signature.
6. Add a Relative Path called `/xmlsecurity` under Default Services, and map it to the policy you just created.
7. Deploy the configuration.

Signing and sending a request

In this section, you use Service Explorer to sign a request to be sent to the policy you just created.

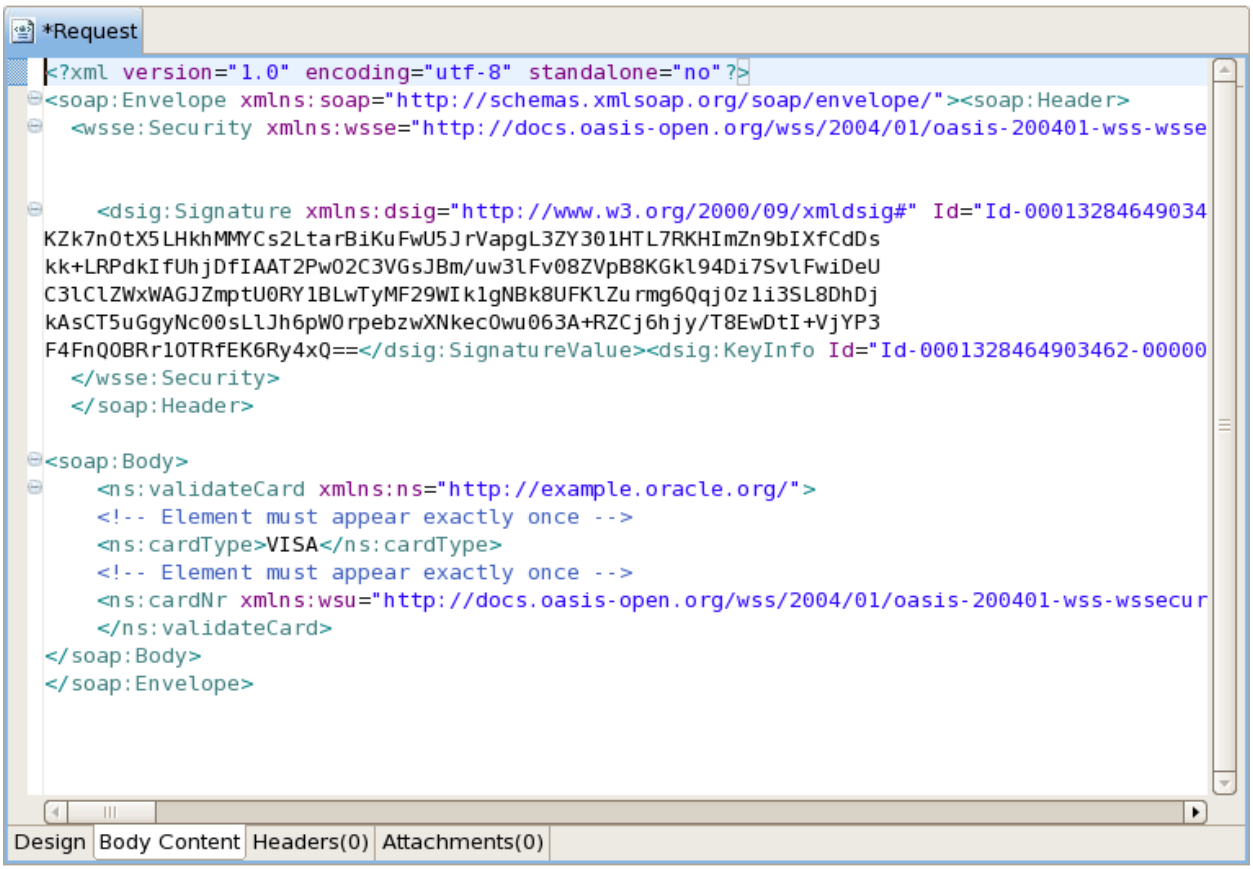
1. Open Service Explorer.
2. Import in Service Explorer the client certificate you exported in Practice 1.

   Select **Security > View Certificates** from the main menu, and perform the steps in the following table:

Step	Window/Page Description	Choices or Values
a.	Select Certificate	Click Create/Import.
b.	Configure Certificate and Private Key	Click Import Certificate + Key.
c.	File Explorer	Navigate to /home/oracle/labs/Lesson_11, select clientCert.p12. Click OK.
d.	Enter password	Enter the password you used when exporting the certificate. Click OK.
e.	Configure Certificate and Private Key	Enter `Client Cert` in the Alias Name field. Click OK.
f.	Select Certificate	Click OK.

3. Create a request from the Request Settings dialog box:
   a. Name this new request ValidatedCC – XMLsecurity.
   b. Make its target URL to be `http://localhost:8080/xmlsecurity`.
4. Load the `ValidateCCRequest.xml` file from /home/oracle/labs/Lesson_11 as request input.
5. Sign the request by performing the following steps:
   a. Select **Security > Sign Request** from the main menu.
   b. Click Signing Key, and select Client Cert, the key you just imported.
   c. On the KeyInfo tab, make sure the “Embed public key information” option is selected.
   d. On the What to Sign tab, repeat the previous steps to create an XPath expression for the CCardNumblock (Step 3.C), and select this XPath to indicate you want to sign the CCardNumblock block.
   e. On the “Where to place the signature” tab, select “In WS Security element” for the current actor/role.

f. Click Finish. The signature appears in the request window resembling the image below:



6. Click the Run button to send a request to the gateway. You should see if the response is the same as the request. This means the verification succeeds.
7. Modify the credit card number, and then send the request. You should get a 500 error response.

   If you change the card type, will the verification fail too?

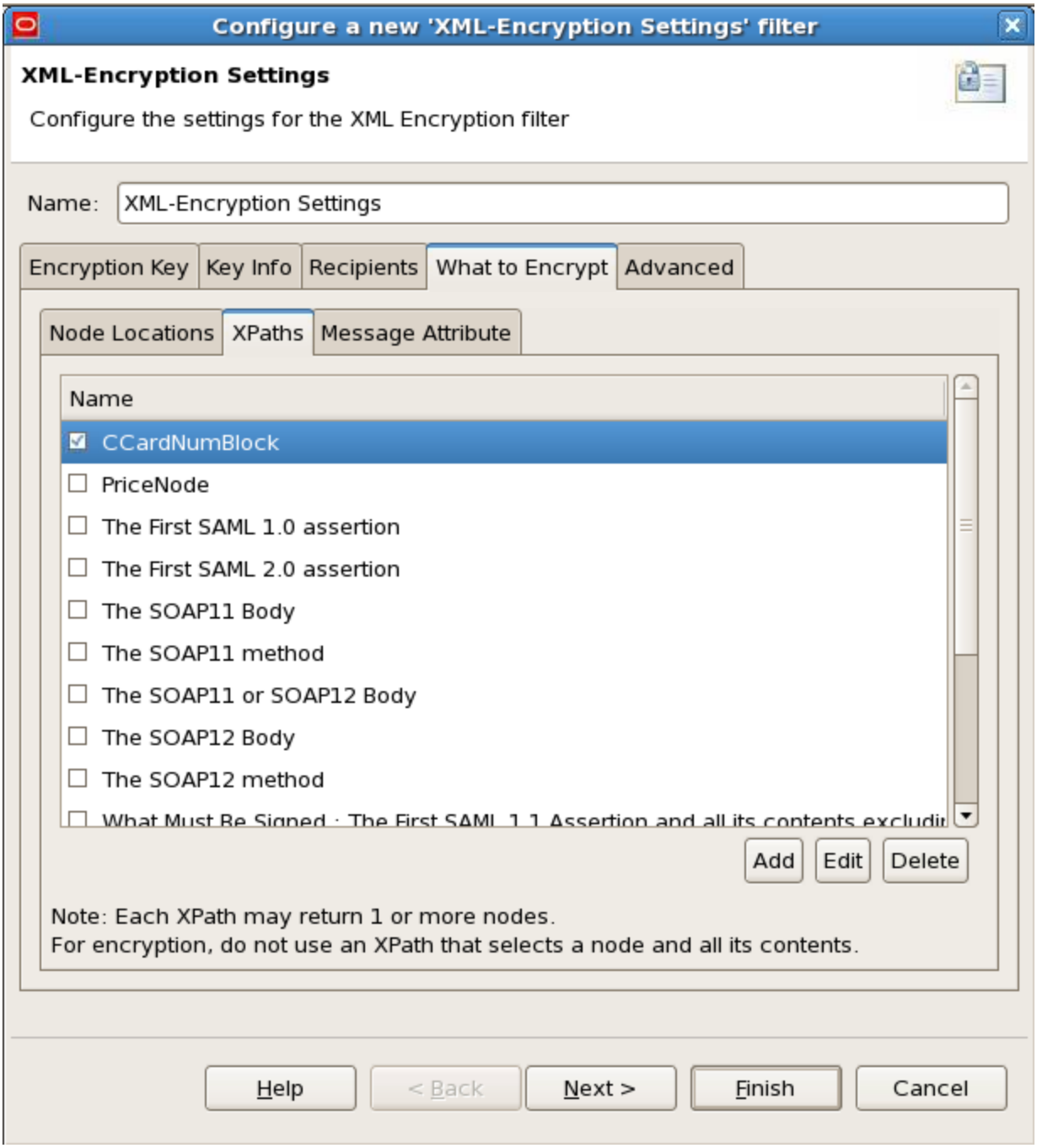
Encrypting the request data

In this section, you enrich the current policy to add encryption. After successful verification of the digital signature, a message attribute containing the incoming certificate is populated. This certificate message attribute can be used to asymmetrically encrypt the symmetric key used to encrypt the data. The symmetric key is automatically generated by the gateway.

To encrypt the incoming message, perform the following steps:

1. In Policy Studio, open the VerifySignature-Encrypt policy you created earlier and drop an XML Encryption Settings filter (from the Encryption category) on the canvas.
   a. In the Configure a new 'XML-Encryption Settings' filter window, click the "What to encrypt" tab.

b. On the XPaths tab, select the CCardNumBlock XPATH expression you created.

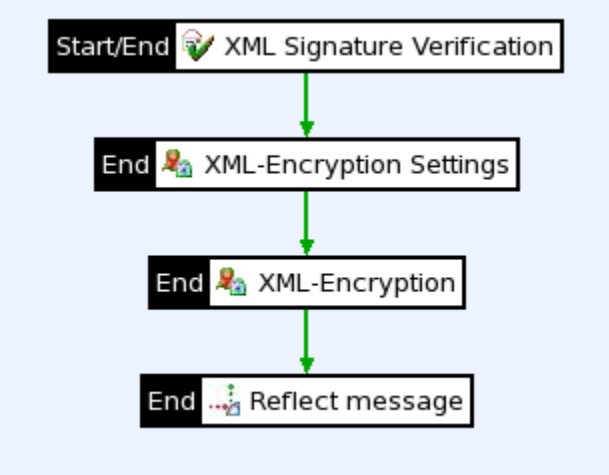


Important Note: If you edit this CCardNumBlock XPath expression, you can change the encryption type and decide whether you want to encrypt the full node (including the actual node name) or the node contents (leaving the node name in clear). Make sure this is set to Encrypt Node.

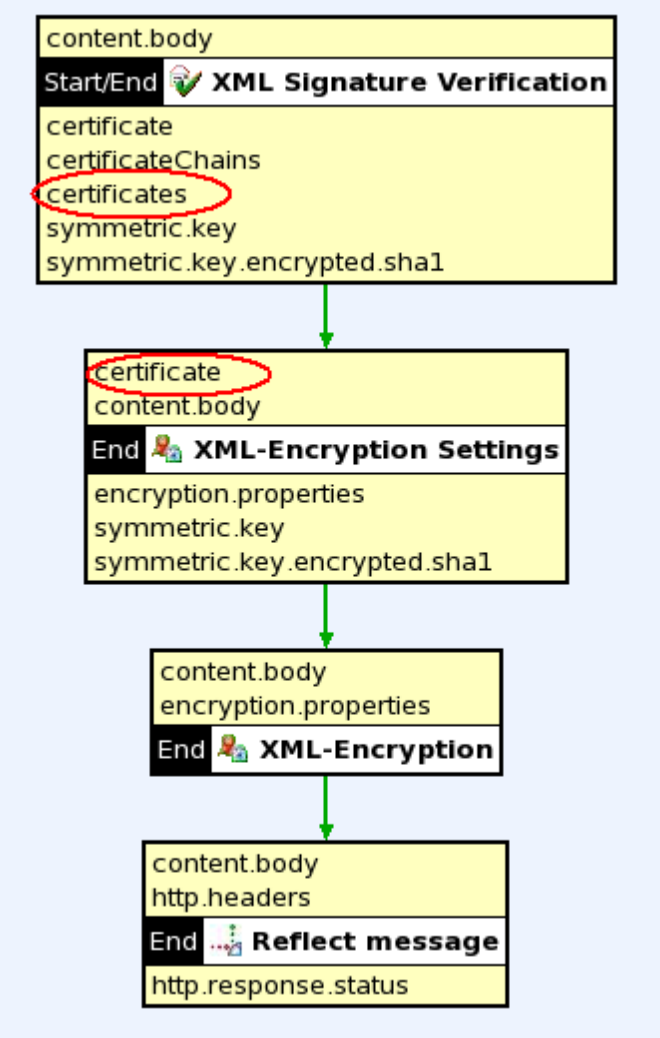
c. Click Finish.

2. Drop an XML Encryption filter right after the XML Encryption settings one. This filter simply executes the encryption based on the previous settings and does not need any configuration.

3. Wire the encryption filters between the signature verification filter and reflect message filter. After it is completed, the policy looks like this:

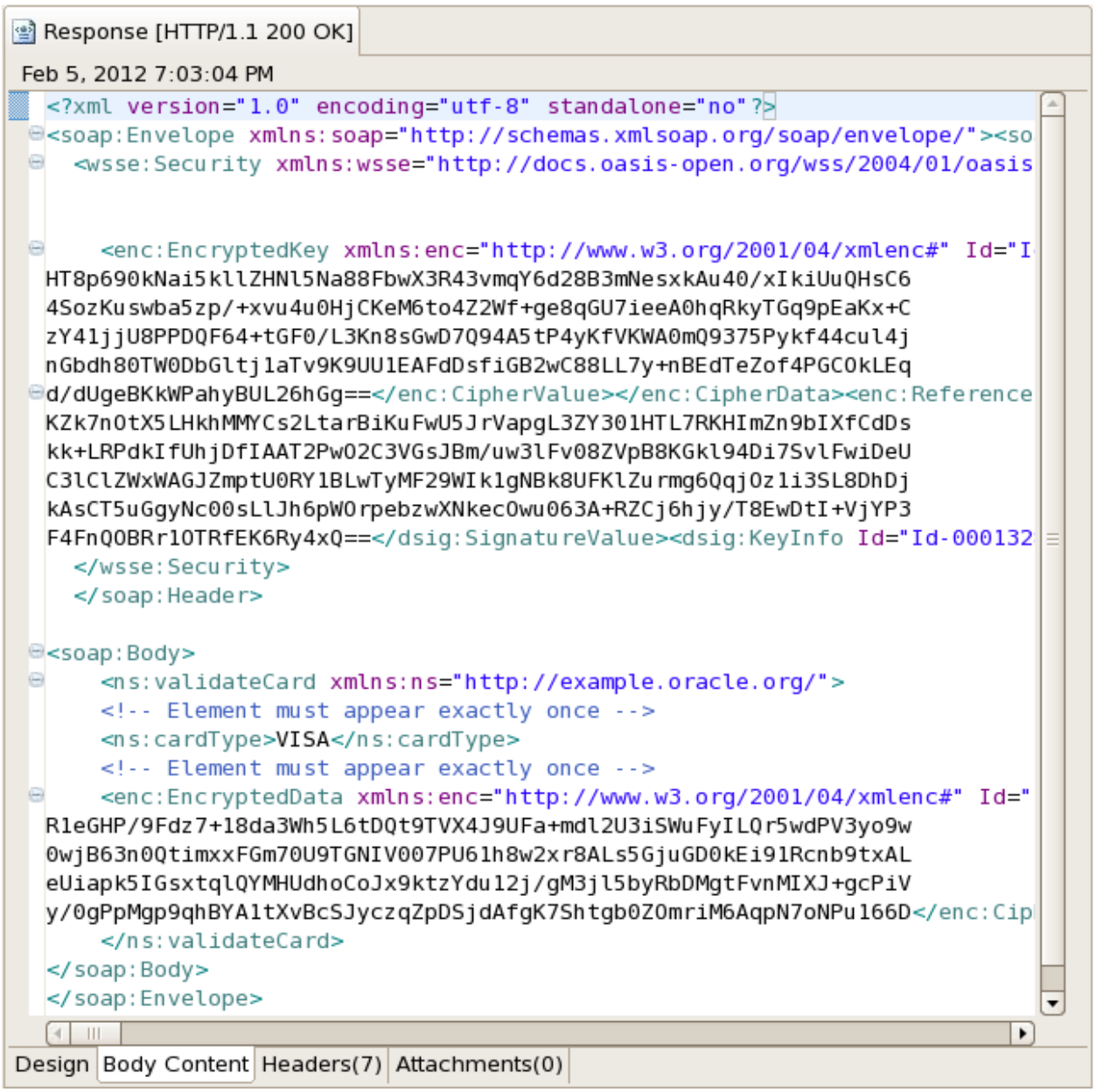


4. Right-click any filter and choose Show All Attributes. This is useful to see how the certificate attribute from the signature verification filter is used in the XML Encryption settings.



5. Deploy the configuration.

6. Go back in Service Explorer and send the request again. You should see if the credit card number is encrypted in the response.



Practice 11-3: Transforming Messages

Overview

In this practice you explore how to use OEG to transform message content by using XSLT to remove sensitive content, in this case a patient name in an HL7 message.

Tasks

The main tasks here are to:

- Import a policy into Policy Studio
- Map the policy to a Relative Path
- Configure a “Stylesheet Conversion” filter

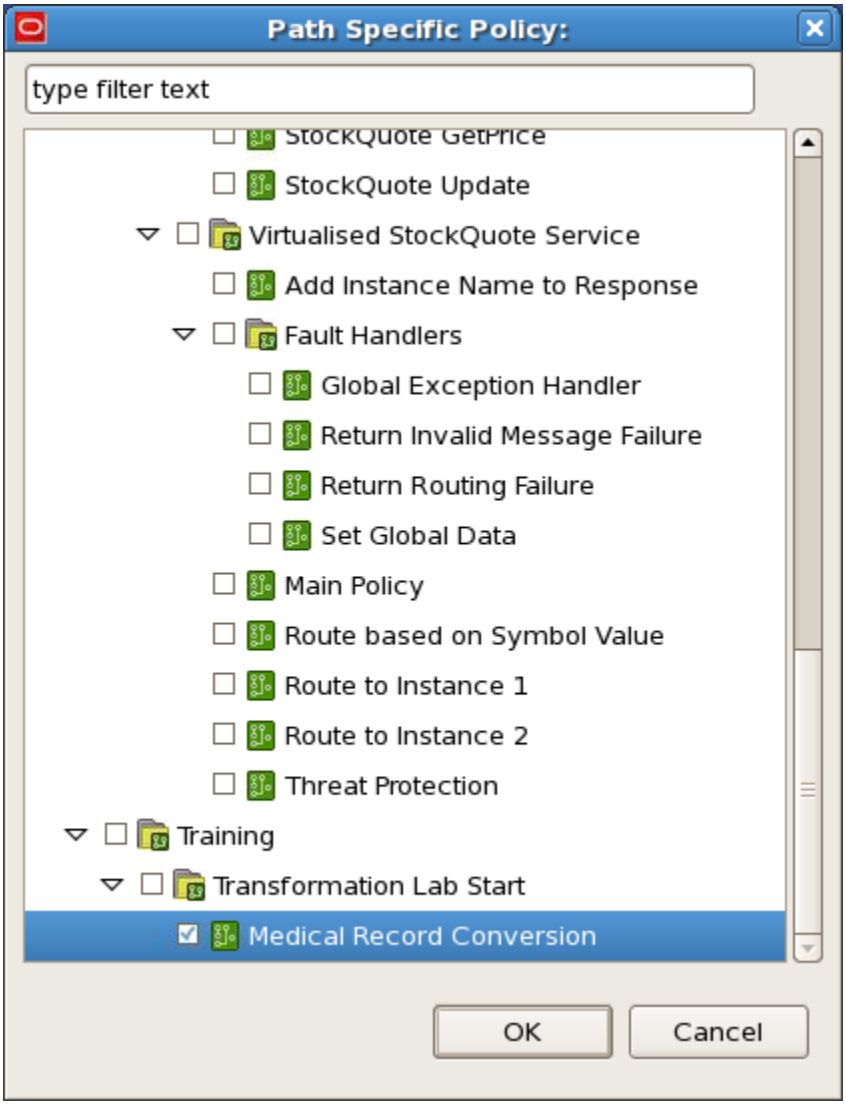
1. Open Policy Studio and import the `conversionPolicy.xml` policy located in the /home/oracle/labs /Lesson_11/Transformation folder.

   After the policy is imported successfully, you should see that a folder named Training is added under the Policies node.

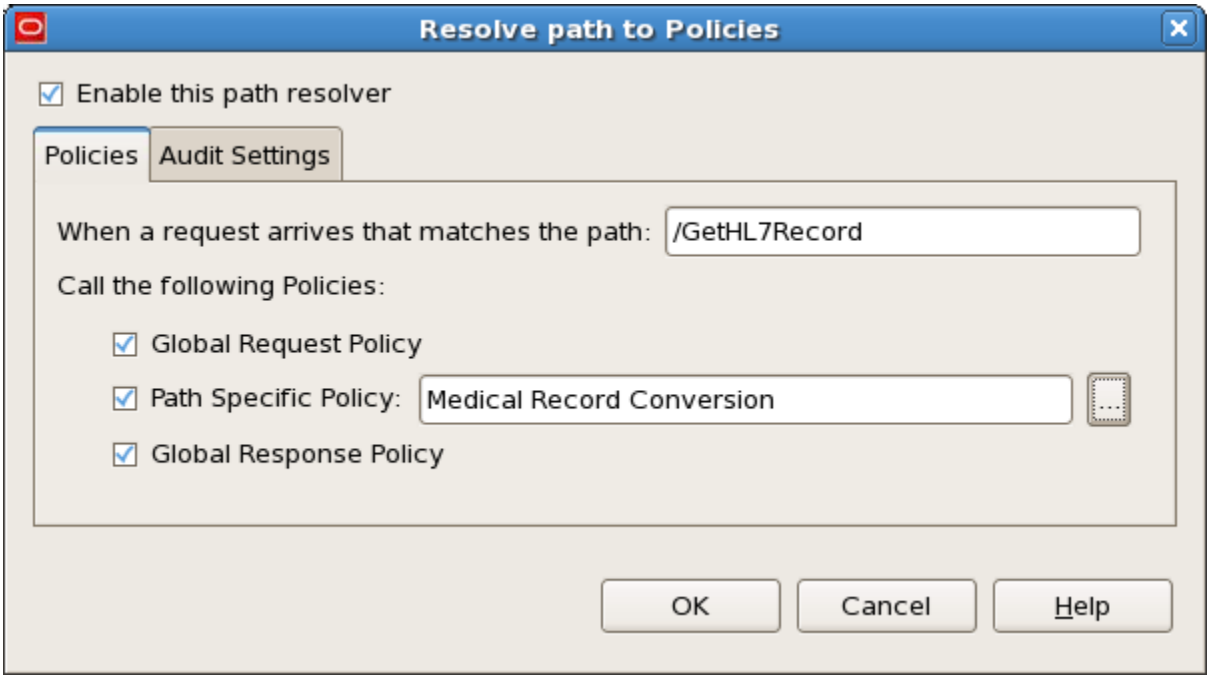
Mapping the policy to a Relative Path

1. Add a new Relative Path, and map the policy to it:
   a. Expand Listeners > Oracle Enterprise Gateway in the left tree pane, right-click the Default Services node, and select “Add Relative Path” from the context menu.
   b. In the Resolve path to Policies window, enter the new Relative Path as shown below (remember to put in the “/” before the path):

c. Click the browsing button next to the Path Specific Policy field, and select the Medical Record Conversion policy.



d. Click OK. The Resolve path to Policies window is updated with the selected policy as shown below:

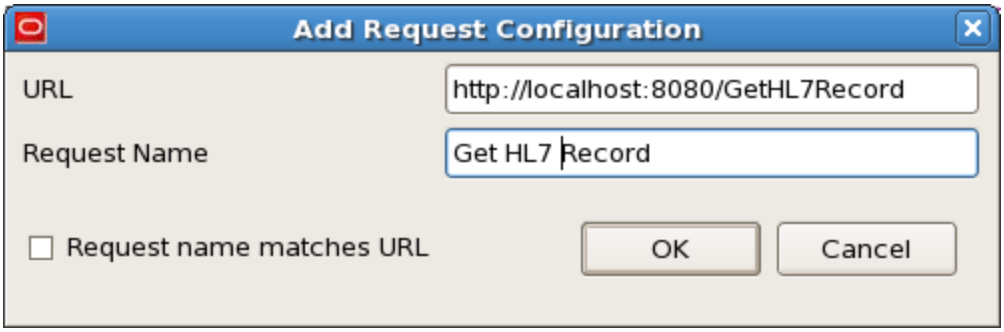


e. Click OK. You should see the `/GetHL7Record` Relative Path added under the Default Services node.

2. Deploy the configuration.

Testing the Relative Path

1. Open Service Explorer, click the inverted triangle in the toolbar, and select "Request Settings…" from the drop-down menu.
2. In the Request Settings window, add a new request.
3. In the Add Request Configuration dialog box, configure the new request as shown below. It will access the Relative Path you just created.



4. Click OK. The "Get HL7 Record" request is added to the request list.
5. Click Close.
6. In Service Explorer, choose the new request you just created.

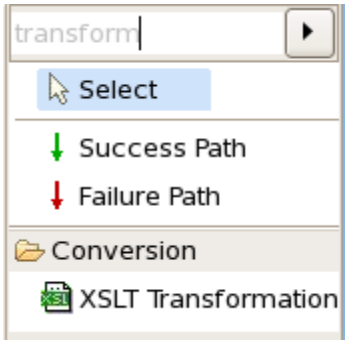
7. Load the `medicalrequest.xml` file from the labs > Lesson_11 > Transformation folder.

8. Click the green “Play” button (in the “Get HL7 Record” request field) to run the request. You should see the response message display in the right pane. It contains private information returned from the service. Next, you will modify the policy to block the personal information with the OEG Gateway.

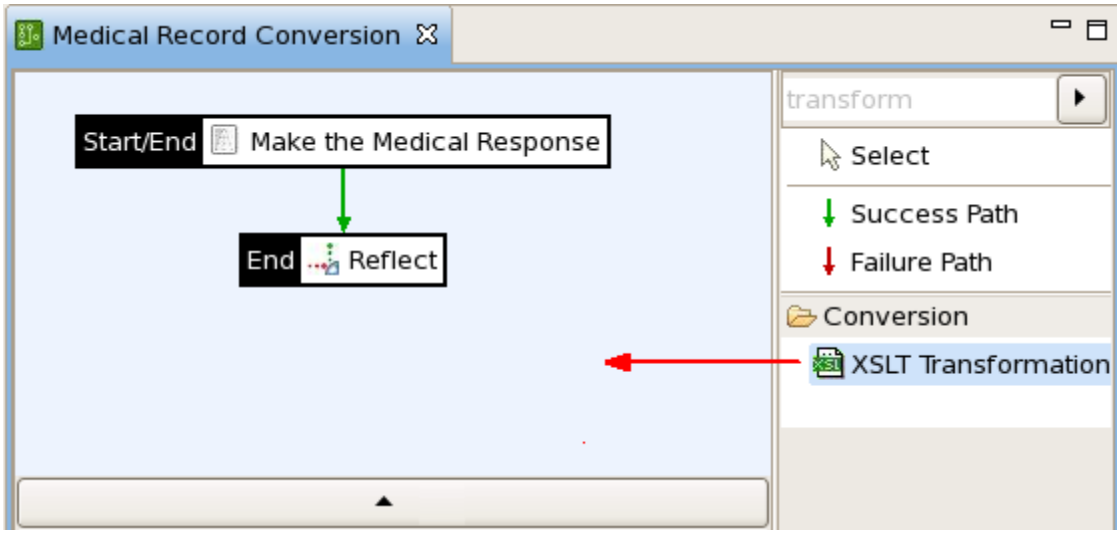
Blocking information by adding a StyleSheet to the policy

1. Open Policy Studio, select Medical Record Conversion policy under Policies > Training > Transformation Lab Start.

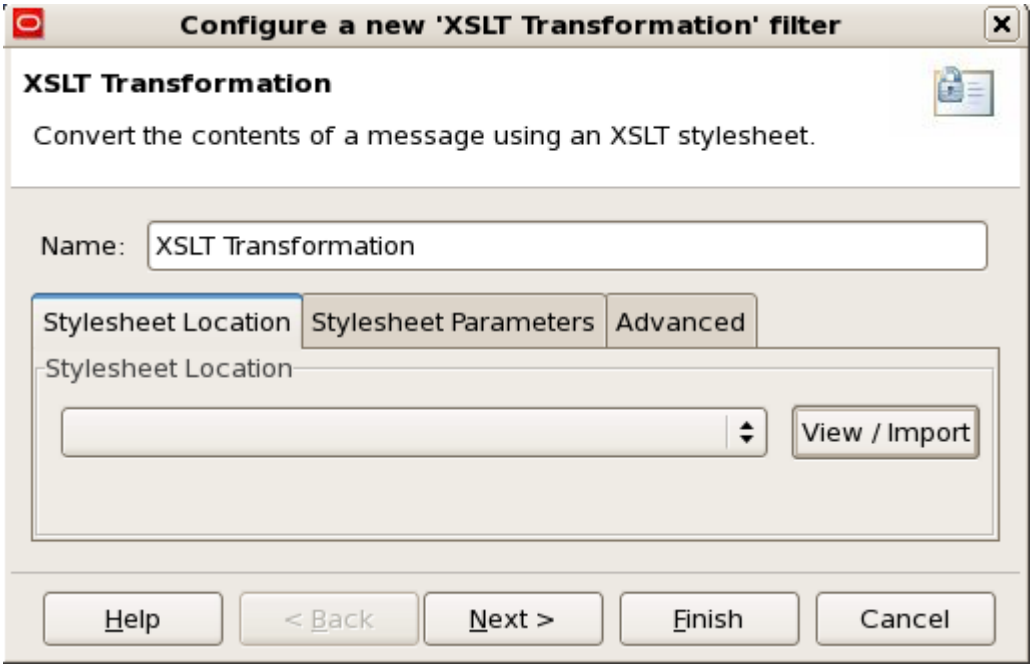
2. In the search field of Filters Palette, enter “transform”, and press Enter to find the stylesheet filter.



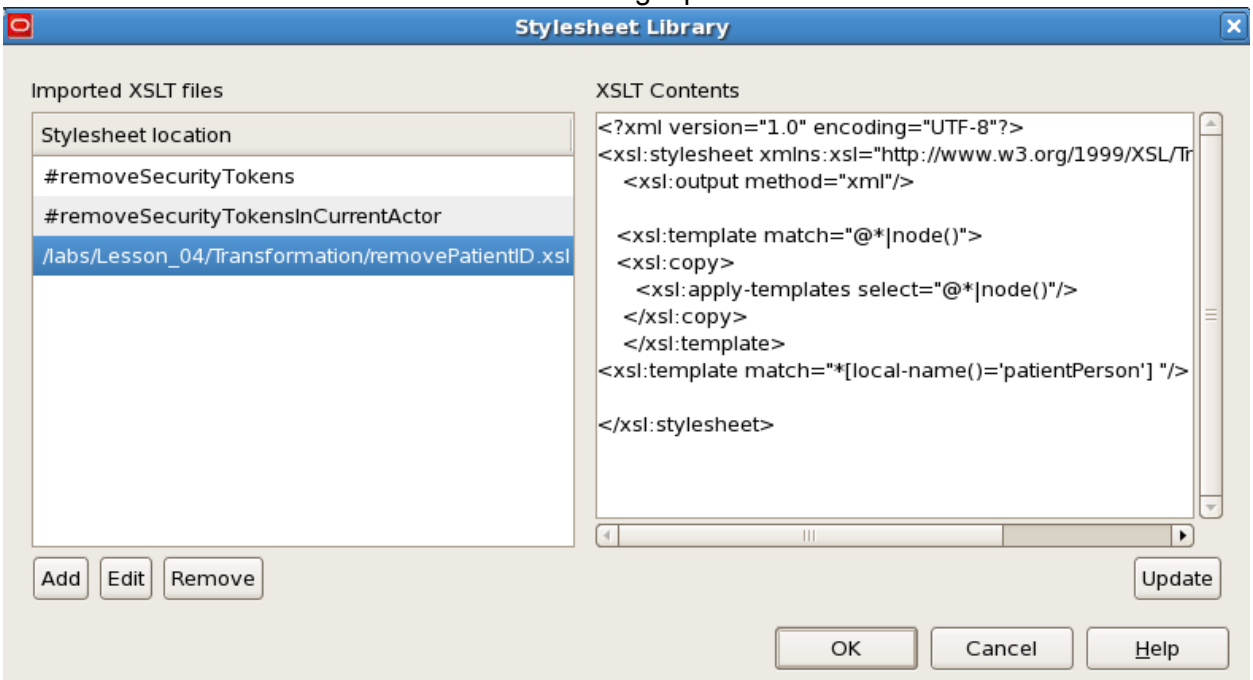
3. Select the XSLT Transformation filter, and drop it on the policy editor canvas. The filter editor opens.



4. In the Configure a new ‘XSLT Transformation’ filter window, under the Stylesheet Location tab, click View / Import.

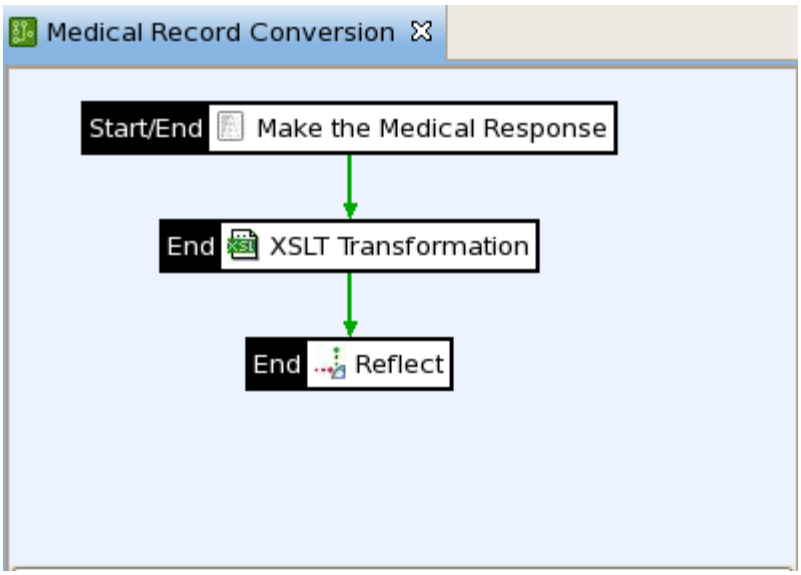


5. In the Stylesheet Library dialog box:
   a. Click Add to open the Browse to Conversion Stylesheet dialog box.
   b. Import the `removePatientID.xsl` file from the labs > Lesson_11 > Transformation folder. You should see the `removePatientID.xsl` file added to the “Imported XSLT files” list. You can view the XSLT content in the right pane.



   Note: The XSLT matches the content of a “patientPerson” element and outputs nothing in response to that element, effectively removing the private information from the message.
   c. Click OK.

6. In the Configure a new ‘XSLT Transformation’ filter window, ensure that the stylesheet you just imported is the one shown in “Stylesheet location”. Click Finish. The XSLT Transformation filter appears as greyed on the editor canvas.
7. “Wire up” the policy so that it looks as shown below:



8. Deploy your changes to the Gateway.

9. Open Service Explorer and send the Medical Request message again. Notice that the patient identifying information is now not returned in the response message.

Practices for Lesson 12: Securing Web Services

Chapter 12

Practices for Lesson 12: Overview

Practices Overview

In these practices, you will learn to create and test the policy that authenticates the user via WSS UsernameToken and secure web services by using WS-Policy.

Practice 12-1: Authenticating the User by using WS-Security Username Token

Overview

In this practice you will secure a web service with OEG by requiring the client to present a valid WSS-Username Token.

Assumptions

The Enterprise Gateway is up and running.

Tasks

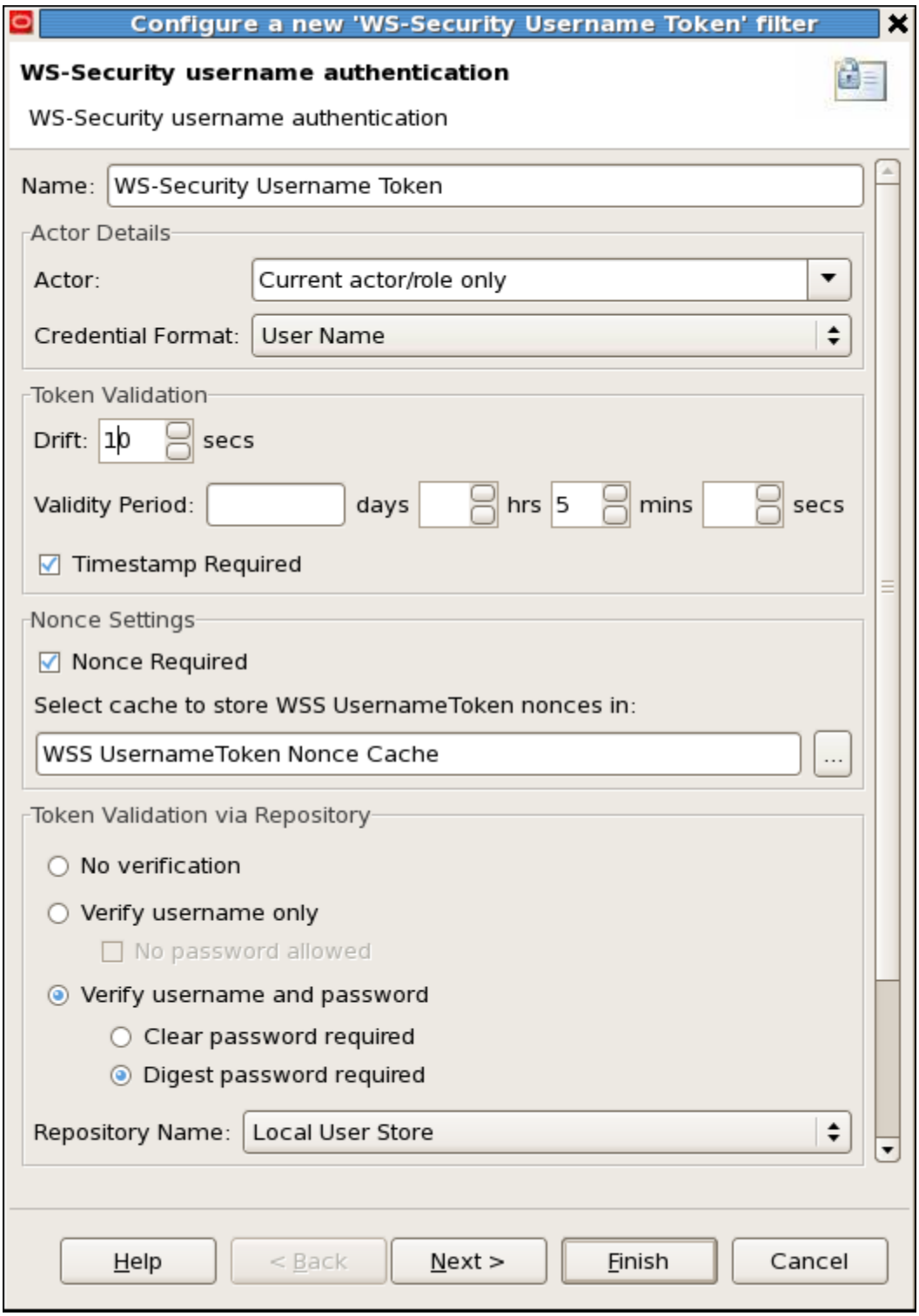
The main tasks you will perform include:

- Using the WS-Security Username to authenticate a user
- Generating a WS-Security UsernameToken from Service Explorer
- Using the WS-Policy wizard to apply a policy to a virtual service

Creating a WSS Username authentication policy

1. Create a new policy called “WSS Username Authentication” in the Policies folder.
2. Drag in a “WS-Security Username Token” filter from the “Authentication” group on the Policy editor canvas.
3. In the Configure a new “WS-Security Username Token” filter window, make the following configuration:
   a. Set the Drift time to 10 seconds.
   b. Set the token Validity Period to 5 minutes.
   c. Make sure Timestamp Required and Nonce Required are selected.

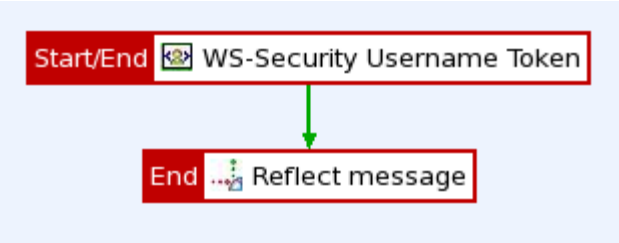
d. By default, the Timestamp and Nonce part of the messages are required. A message without this information would not be accepted.
e. Select Digest password required to verify username and password.
f. Select Local User Store as the repository name.



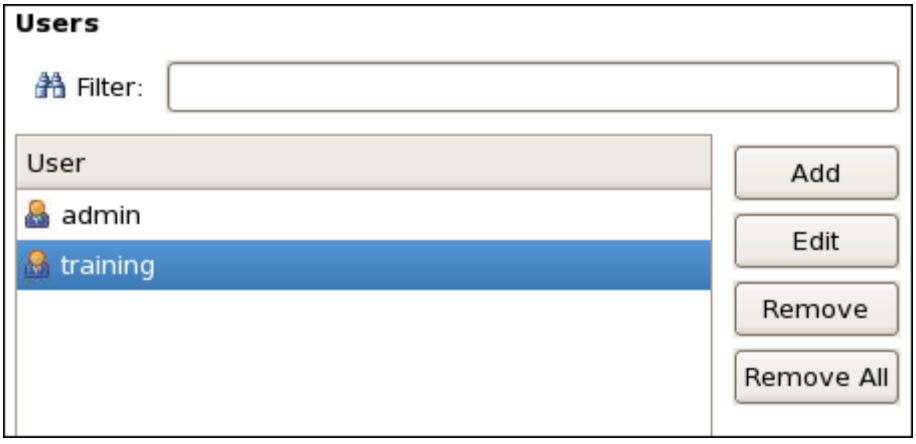
g. Click Finish.

4. Right-click the filter, and select Set as Start.

5. Drag in a “Reflect message” filter and place it after the WS-Security Authentication filter.



6. Create a user in the Local User Store and perform the following steps:
   a. Navigate to Users And Groups > Users.
   b. Add a user with User Name: `training`, and Password: `oeg`.



7. Navigate to Default Services node, add a Relative Path called “/Authentication_UsernameToken” and map it to the “WSS Username Authentication” policy.
8. Deploy the configuration.

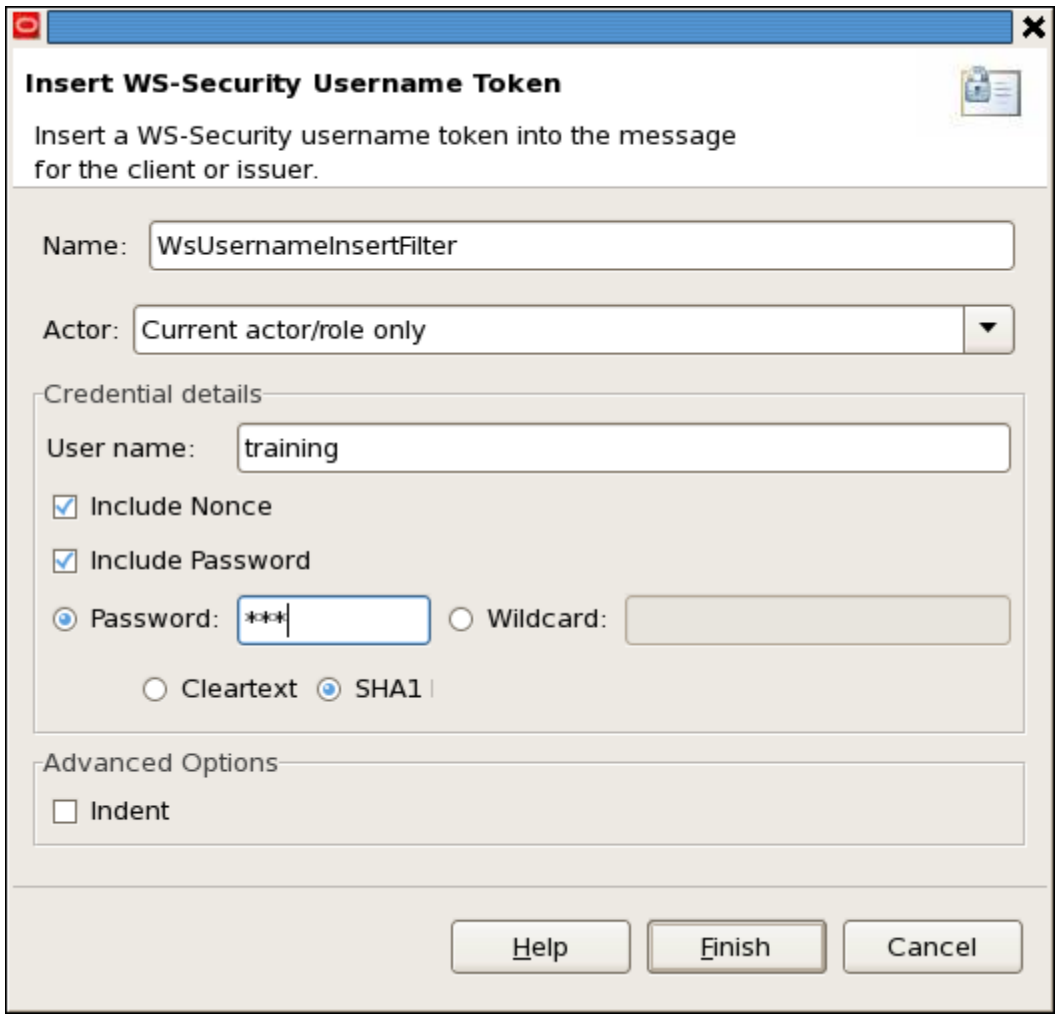
Testing the policy

1. In Service Explorer, open a sample SOAP message shipped with Service Explorer by selecting File > Samples > Add two numbers from the main menu.

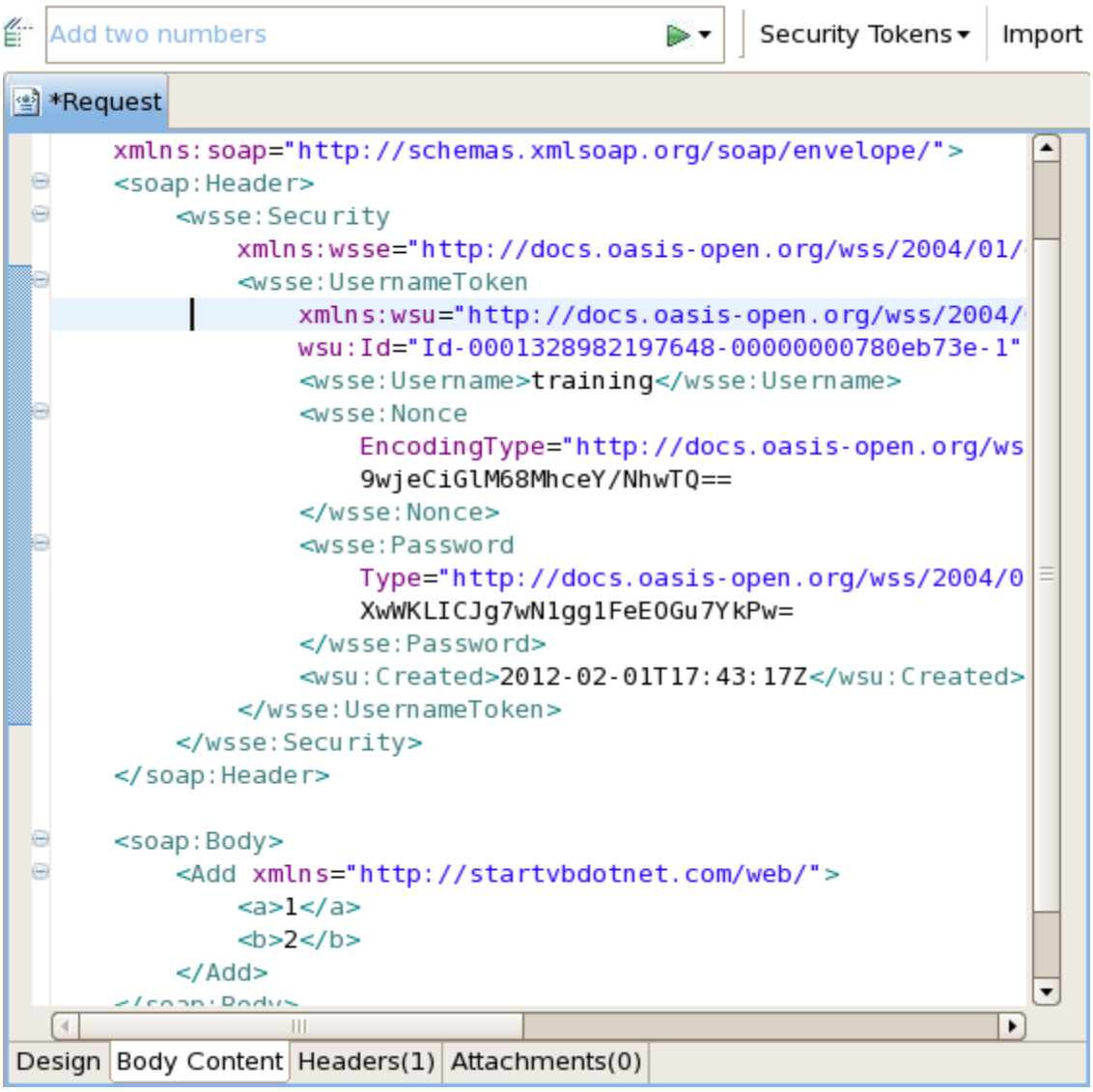
2. Modify its target to point to the Relative Path you just created:
   a. Select Request Settings from the tool bar.

   b. In the Request Settings window, change the URL to `http://localhost:8080/Authentication_UsernameToken`.
3. Click Run to send the SOAP message to the service. Without the UsernameToken, you should see the message is blocked.
4. To insert the Username Token in the request message, perform the following steps:
   a. Click “Security Tokens,” and select “Insert WS-Security Username.”

b. Enter the details to match the user you have created. Use the following image as a guide:



5. You should see the request SOAP message is updated with the Username Token information in the SOAP header.



6. Send the request. With the Username Token in place, the message is passed.
7. If the message is sent a second time, it is blocked as a replay attack.

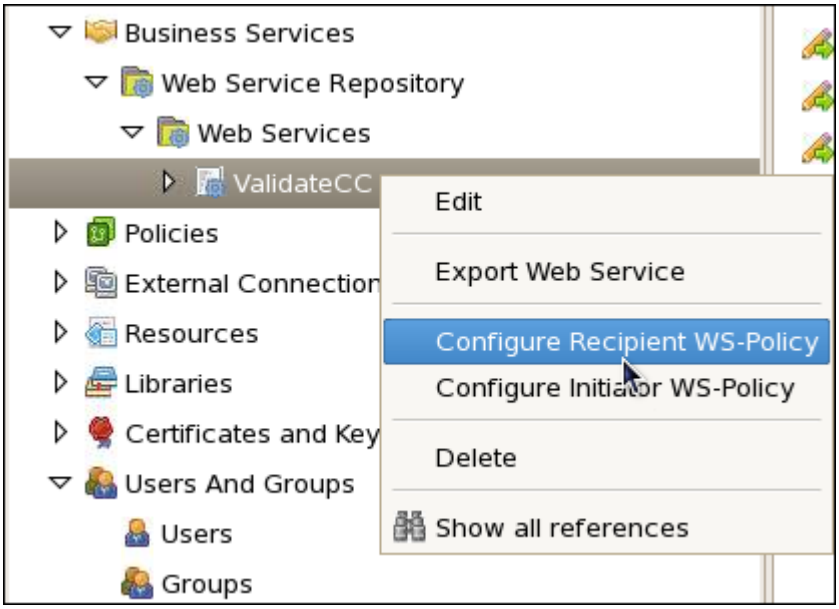
Practice 12-2: Securing a Service by Using the WS-Policy

Overview

In this practice you use a specific wizard to apply a standard WS-Policy and generate the corresponding WSDL. Service clients can then consume this policy and enforce it in their code.

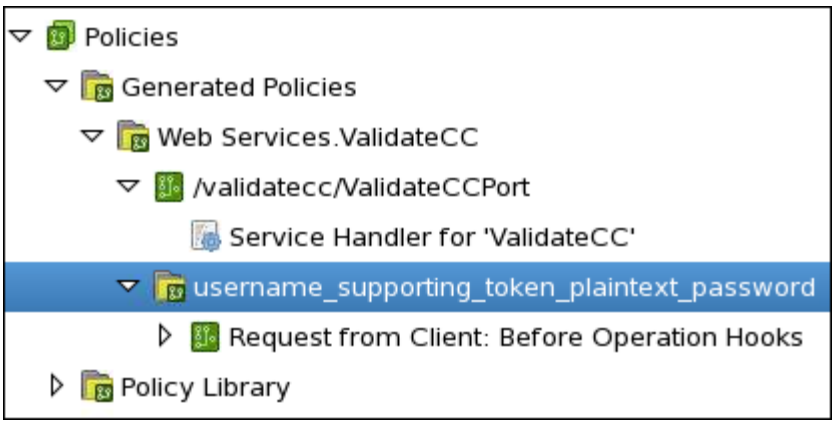
Tasks

1. In Policy Studio, navigate to Business Services > Web Service Repository > Web Services.
2. Right-click the ValidateCC node and invoke Configure Recipient WS-Policy.



3. In the “Secure Virtual Service” dialog box, click the Gateway policy drop down, and select “Username Supporting Token Plaintext Password”.
4. Click OK.
5. In the “Configure Recipient Security Settings” dialog box, select “Validate Client’s WS-Security UsernameToken” in the left frame, and configure the settings in the right as follows:
   a. Set the Drift time to 10 seconds.
   b. Set the token Validity Period to five minutes.
   c. Check the Timestamp Required and Nonce Required options.
   d. Make sure Clear password required is selected, and Local User Store is selected as Repository Name.
6. Click Finish.

7. Look inside the Generated Policies container to understand what has been generated and how it is used from the Service Handler filter.



8. Deploy the configuration.

Testing in Service Explorer

1. Reuse the request you had created for the Lab 4: ValidateCC - validateCard and test this new service from Service Explorer without a username token, and view the response telling you the message is blocked.
2. Insert a Username token with plaintext password (no Digest), and send the request again. With the Username Token in place, you should get a return of the validation result.

3. Open Firefox browser, and enter the following URL:

```
http://localhost:8080/validatecc/ValidateCCPort?WSDL
```

You can view the WSDL of the virtual service enforced by a WS-Policy that requests the client to present a Username Token to access the service.

Practices for Lesson 13: Securing SOA Composites with OEG and OWSM

Chapter 13

Practices for Lesson 13: Overview

Practices Overview

In these practices, you will get a feel of how to secure a SOA composite application in a multi-tier deployment architecture with OEG, OSB, SOA, and OWSM.

The back-end application/web service is applied with a WLS security policy that authenticates a SOAP request by using WS-UsernameToken, but the client request (sending through Service Explorer) uses HTTP basic authentication. So you will also learn to use policy at the Gateway level to convert HTTP Basic authentication to WS-Security Username Token.

Practice 13-1: Deploying and Examining the SOA application

Overview

In this practice you deploy a SOA composite application, a credit card validation application, to the managed SOA server. Similar to the ValidateCC web service you used in the previous labs, this application also takes a credit card number and returns the status, that is, VALID or INVALID. In this example, the only valid Credit Card number is 1234-1234-1234-1234.

Tasks

As SOA and OSB are memory hungry programs, make sure you shut down all the unnecessary applications before starting this lab. At this point, only keep the following programs up and running:

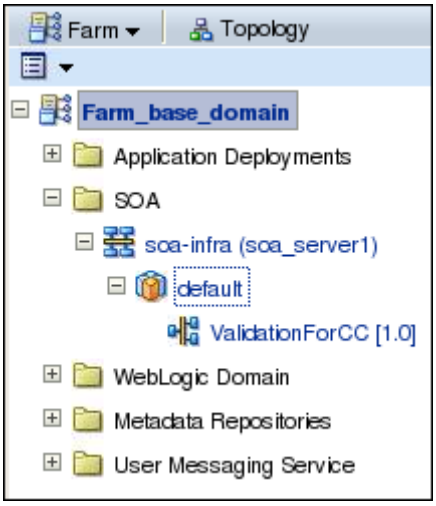
- Weblogic Admin Server
- Enterprise Gateway
- Policy Studio

1. To start the SOA managed server, perform the following steps:
   a. Locate the “Start SOA Server” icon on the Desktop, double-click it.
   b. After a short time, you are prompted for the administration credentials. Enter the username and password as shown:
      ```
      Enter username to boot WebLogic server: weblogic
      Enter password to boot WebLogic server: welcome1
      ```
   c. It takes a few minutes for the server to start up, so wait until you see a message similar to:
      ```
      <Dec 15, 2011 12:30:13 PM UTC> <Notice> <WebLogicServer> <BEA-
      000360> <Server started in RUNNING mode>
      SOA Platform is running and accepting requests
      ```
   d. Minimize the terminal window.
2. Deploy the sample SOA composite application by using command line:
   a. Open a terminal window.
   b. Go to the /home/oracle/labs/Lesson_13 directory, and list the files:
      ```
      [oracle@EDCPR16P0 ~]$ cd labs/Lesson_13
      [oracle@EDCPR16P0 ~]$ ls –l
      ```
      You should see the following files listed:
      ```
      authentication_SOA.xml
      deploy_soa.sh
      sbconfig.jar
      sca_ValidationForCC_rev1.0.jar
      ValidationForCC_osbconfig.jar
      ```
   c. Run the deploy command:
      ```
      $./deploy_soa.sh sca_ValidationForCC_rev1.0.jar
      ```

You should see the following message if the deployment succeeds:

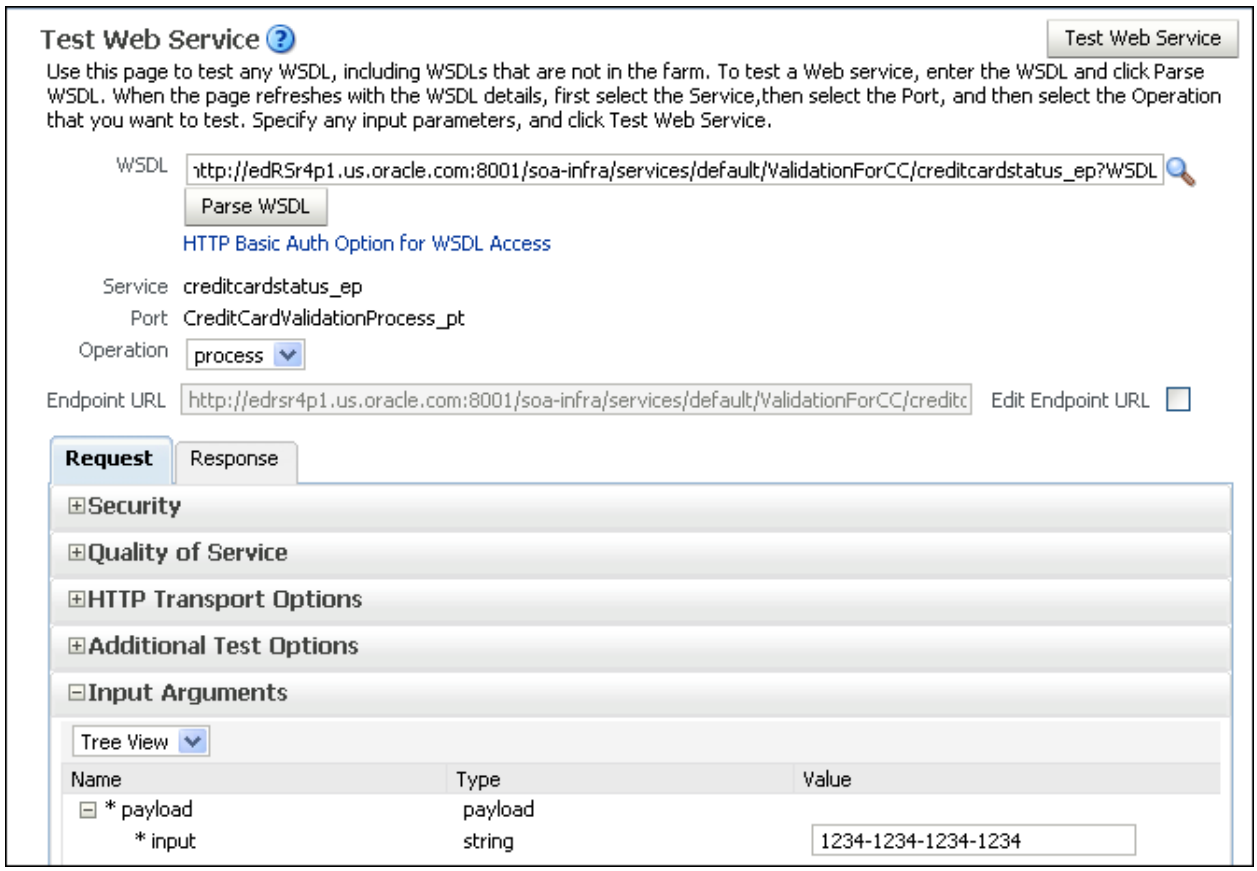
```
Buildfile: /u01/app/oracle/fmw/11.1.1.5/Oracle_SOA1/bin/ant-sca-deploy.xml
     [echo] oracle.home = /u01/app/oracle/fmw/11.1.1.5/Oracle_SOA1/bin/..

deploy:
    [input] skipping input as property serverURL has already been set.
    [input] skipping input as property sarLocation has already been set.
[deployComposite] setting user/password..., user=weblogic
[deployComposite] Processing sar=sca_ValidationForCC_rev1.0.jar
[deployComposite] Adding sar file - /home/oracle/labs/Lesson_12/sca_ValidationForCC_rev1.0.jar
[deployComposite] INFO: Creating HTTP connection to host:localhost, port:8001
[deployComposite] INFO: Received HTTP response from the server, response code=200
[deployComposite] ---->Deploying composite success.

BUILD SUCCESSFUL
Total time: 6 seconds
```

3. Verify the deployment in Enterprise Manager.
   a. Open the Firefox browser, start Enterprise Manager at:
      `http://localhost:7001/em`
   b. Log in as the administration user `weblogic` and associated password (`welcome1`).
   c. On the Farm home page, in the navigation frame, expand the SOA > soa-infra > default node, you should see the deployed application - ValidationForCC [1.0].



   d. Click the ValidationForCC [1.0] link, you can view the application information in the ValidationForCC [1.0] > Dashboard tab page.
      This application contains a BPEL component, which is exposed as a web service, creditcardstatus_ep. This is the entry point to the composite application.
4. You can run some tests to see how this application works.
   a. On the ValidationForCC [1.0] home page, click Test.
   b. On the Test Web Service page, accept all default settings, and then provide an input string for the test. Do the following:
      1) Make sure the Request tab is selected, and scroll down to locate the Input Arguments section.

2) With the default Tree View mode, in the payload > input field, enter the value: `1234-1234-1234-1234`, and click Test Web Service. Use the following image as a guide:



c. On the Test Web Service page, the page is refreshed with the Response tab becoming active and showing the test result as VALID (credit card status).

d. Similarly, execute the test case with a different credit card number, for example 1234-1234-1234-0000, and verify that the result field shows the INVALID credit card status.

Practice 13-2: Virtualizing the web service in OSB

Overview

In this practice you virtualize the ValidationForCC application (exposed as a web service) in OSB server.

The virtualized service configuration has already been created for you, and exported as a JAR file. So in this practice, you just need to import it into OSB.

Tasks

1. To start the managed OSB server, perform the following steps:
   a. Locate the “Start OSB Server” icon on the Desktop and double-click it.
   b. After a short time, you are prompted for the administration credentials. Enter the username and password as shown:

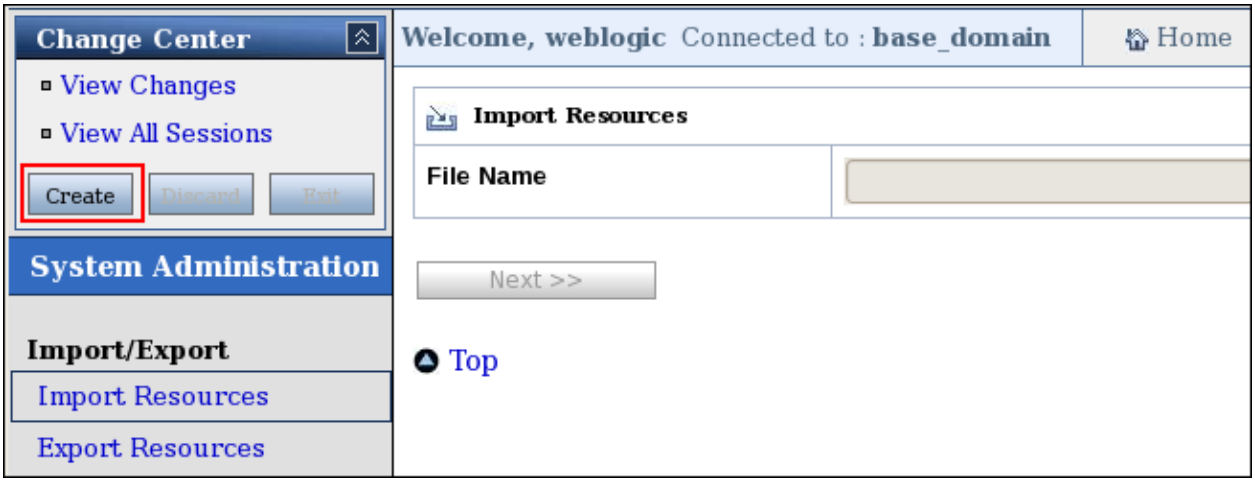
      ```
      Enter username to boot WebLogic server: weblogic
      Enter password to boot WebLogic server: welcome1
      ```

   c. It takes a few minutes for the server to start up, so wait until you see a message similar to:

      ```
      <Dec 15, 2011 1:00:28 PM UTC> <Notice> <WebLogicServer> <BEA-
      000360> <Server started in RUNNING mode>
      ```

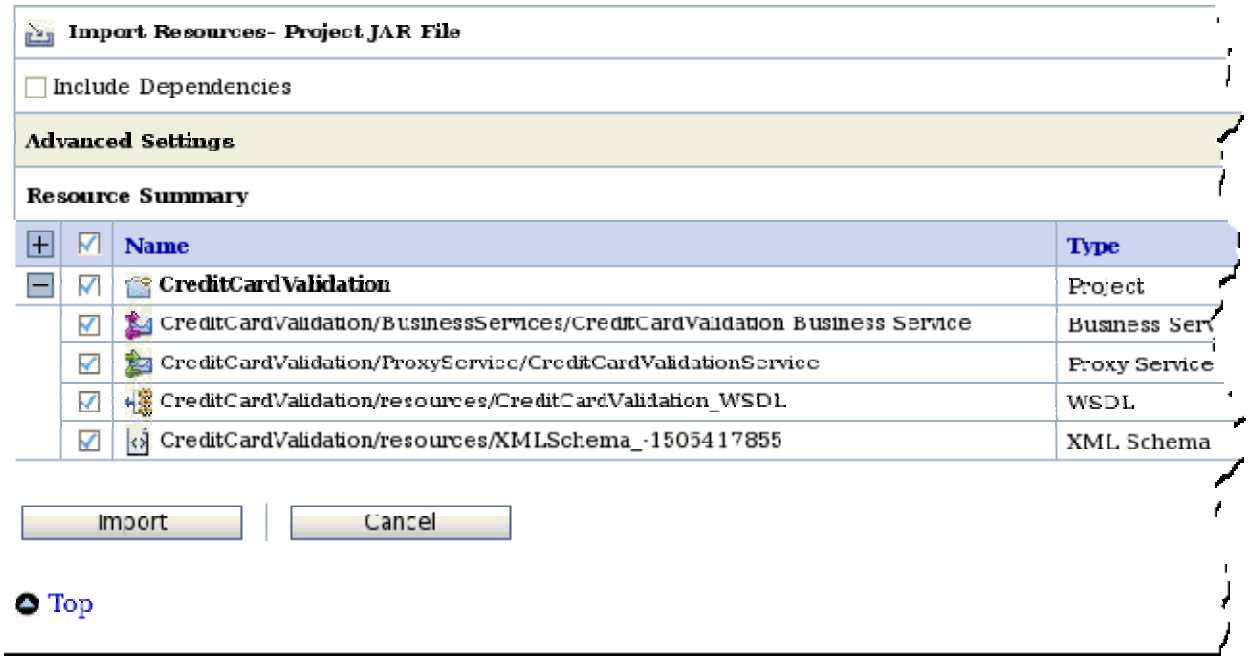
   d. Minimize the terminal window.
2. Open the Firefox browser, start the Service Bus Console at `http://localhost:7001/sbconsole`, and log in as the administration user with `weblogic / welcome1`.
3. Import the OSB project that includes all the configurations for the virtualized web service. Perform the following steps:
   a. In the left navigation frame, scroll down and click the System Administration link.

b. In OSB console, all configuration changes are made in sessions that can be activated or discarded. So before you make any changes, you need to start a new session. Click the “Create” button in the change center at the top left corner of the console.



c. Click the Browse button, and import the `ValidationForCC_osbconfig.jar` file from the /home/oracle/labs/Lesson_13 folder.

d. Click Next. The Import Resource page displays the files included in the OSB project JAR file.

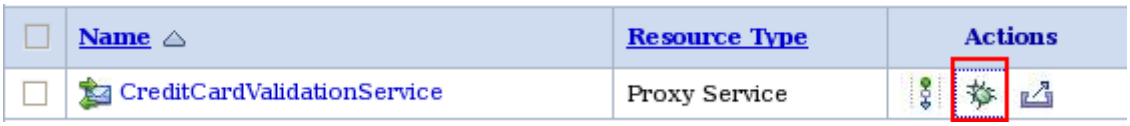


e. Click Import. Wait a few seconds and you should see an import successful message.

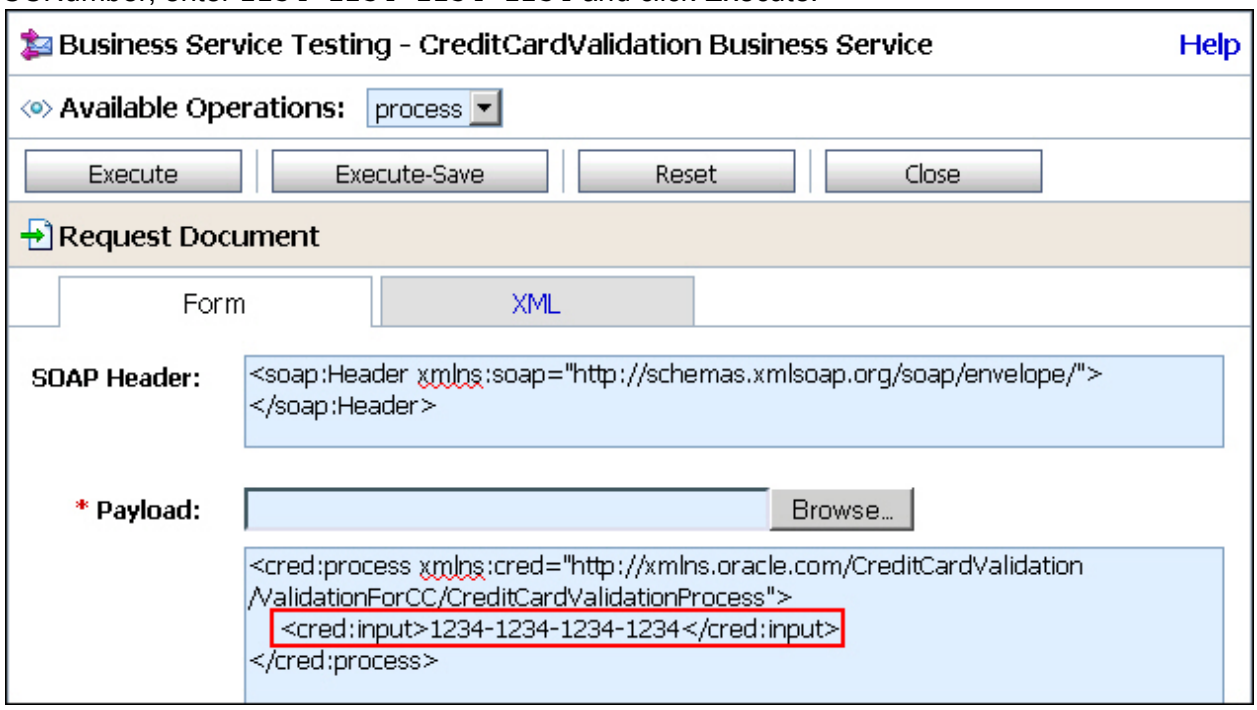
4. Activate the changes made in the current session:
   a. Click the Activate button in the Change Center in the upper left hand corner.
   b. In the Activate Session page, OSB presents a form into which change notes can be recorded for future reference. You can add a note for this session (optional), and click the Submit button.
5. To examine the configuration, perform the following steps:
   a. Click the Project Explorer link at the bottom of the navigation frame. You should see the CreditCardValidation folder is displayed under Projects.

b. Click the CreditCardValidation folder and three nodes are displayed:
   - ProxyService – This is the service exposed by OSB to OEG.
   - BusinessServices – This service represents the back-end service.
   - Resources – This contains the registered WSDL of the back-end service that you will use in the business service.

6. Test to see if the virtualized web service works, perform the following steps:
   a. Click the ProxyService link.
   b. On the CreditCardValidation/ProxyServices page, the actions column has a bug-like icon; click that icon to launch the Test Console for the CreditCardValidation Proxy Service.

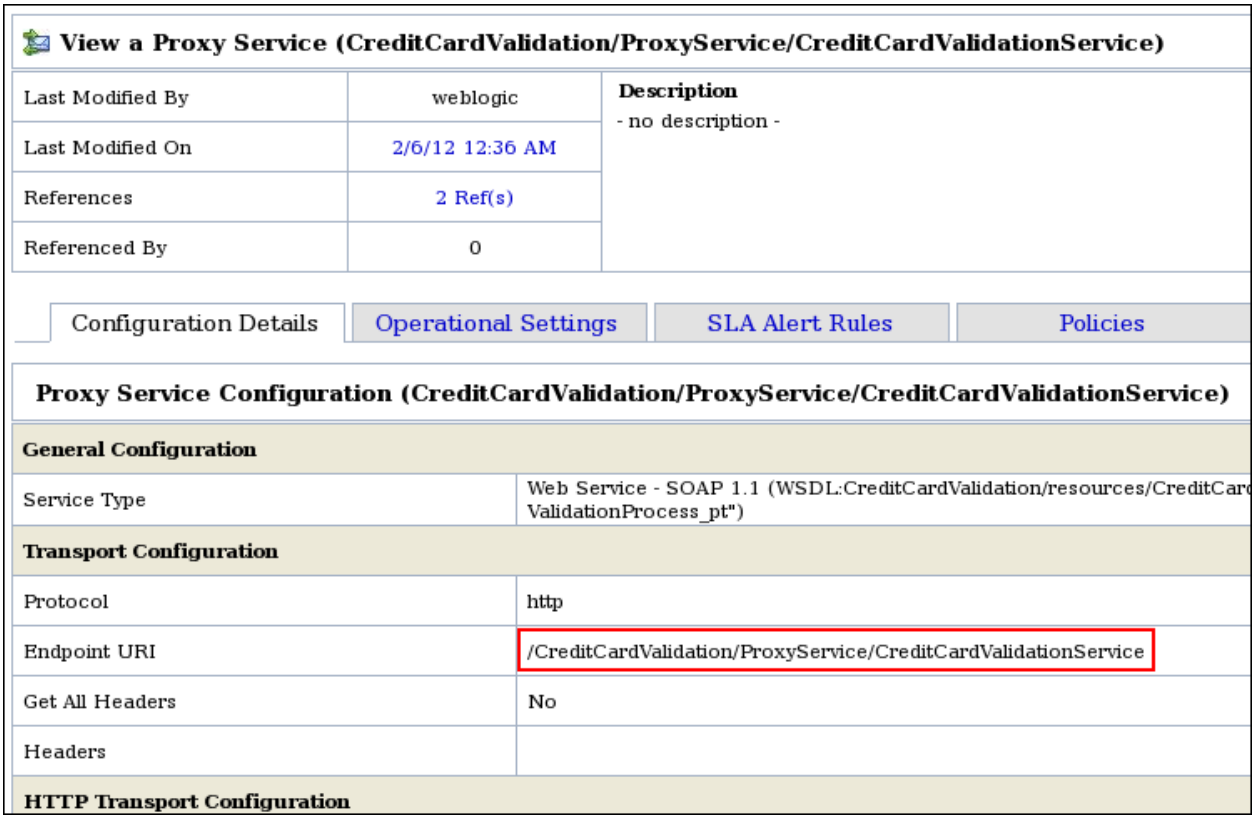


   c. The Test Console window is displayed. You should see a prepopulated XML document, based on the XSD for the request message. In the payload data, for CCNumber, enter `1234-1234-1234-1234` and click Execute.



7. Test again if you wish, with an invalid number (123-123).
8. Close the Test Console window when you are done.

9. Back in the CreditCardValidation/ProxyServices page, click the CreditCardValidationService link. The page shows the proxy service information.



View a Proxy Service (CreditCardValidation/ProxyService/CreditCardValidationService)

Last Modified By	weblogic
Last Modified On	2/6/12 12:36 AM
References	2 Ref(s)
Referenced By	0

Description
- no description -

Configuration Details | Operational Settings | SLA Alert Rules | Policies

Proxy Service Configuration (CreditCardValidation/ProxyService/CreditCardValidationService)

General Configuration	
Service Type	Web Service - SOAP 1.1 (WSDL:CreditCardValidation/resources/CreditCard ValidationProcess_pt")
Transport Configuration	
Protocol	http
Endpoint URI	/CreditCardValidation/ProxyService/CreditCardValidationService
Get All Headers	No
Headers	
HTTP Transport Configuration	

The wsdl of the proxy service can be accessed by using the following URL:
http://localhost:8011/CreditCardValidation/ProxyService/CreditCardValidationService?wsdl
The default port for the OSB server is 8011.

10. Now the back-end service (on SOA server) is properly virtualized in the OSB server.

Practice 13-3: Registering the web service in OEG

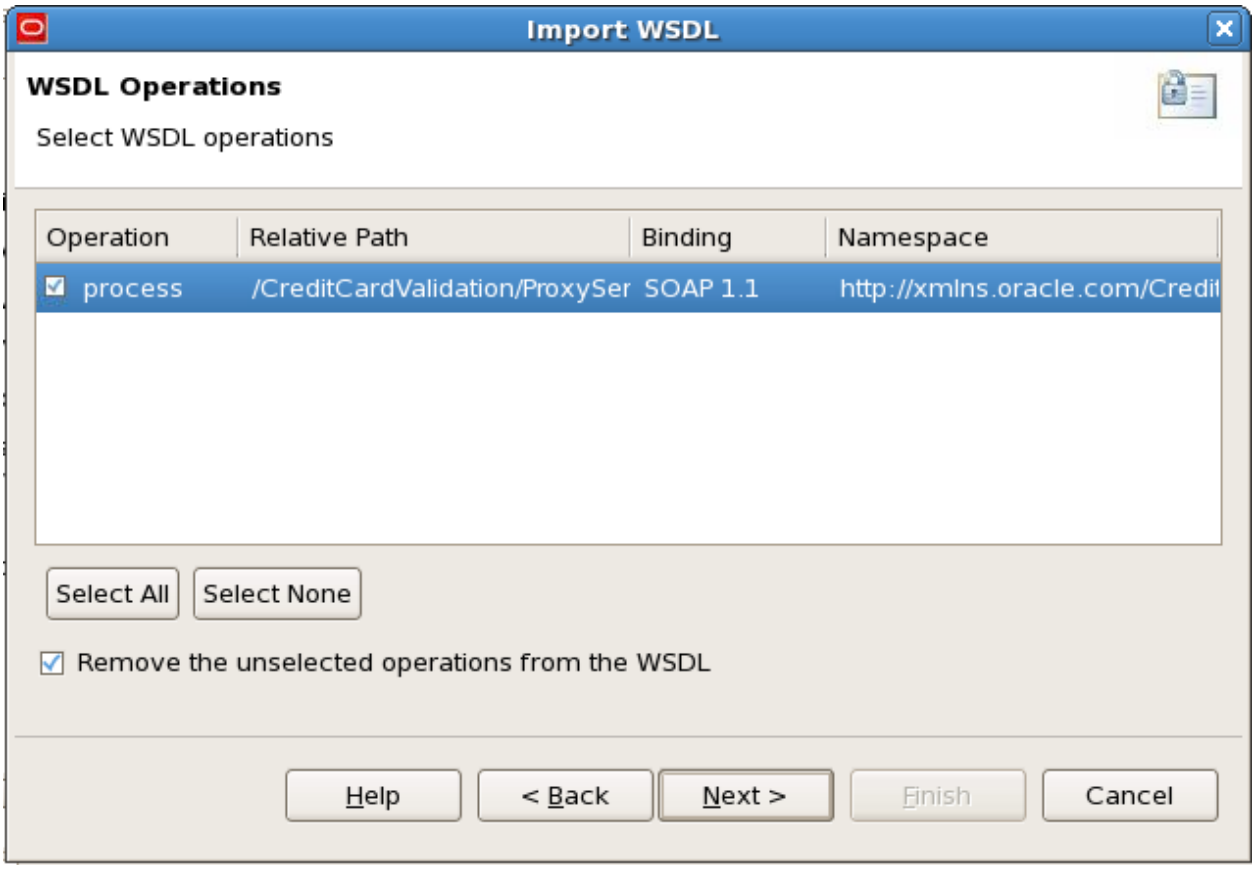
Overview

In this practice you register the proxy service exposed by OSB in the Enterprise Gateway.

Tasks

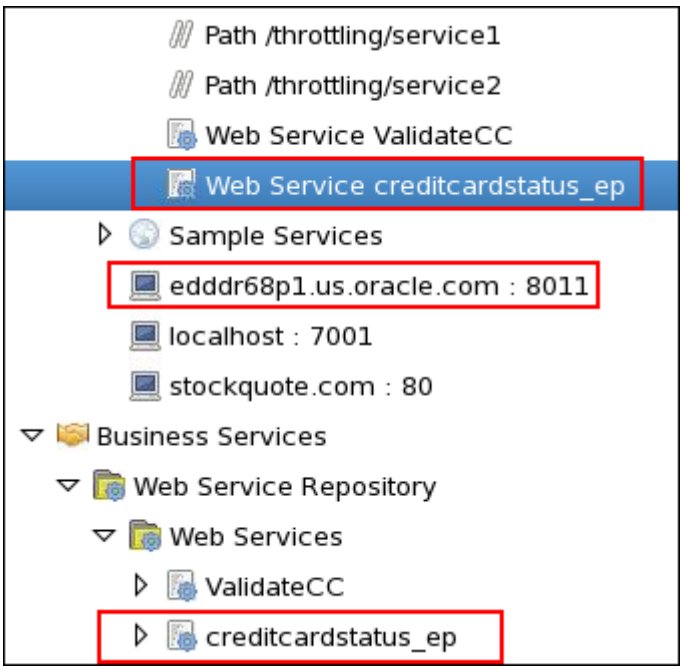
1. Open Policy Studio, expand the Business Services > Web Service Repository tree node, right-click Web Services, and select Register Web Service from the drop-down menu. The WSDL Import wizard opens.
2. In the Import WSDL wizard, on the Load WSDL screen, select WSDL URL option for WSDL Location, and use the URL of the proxy service on OSB:

   ```
   http://localhost:8011/CreditCardValidation/ProxyService/CreditCardValidationService?wsdl
   ```
3. Click Next after entering the URL.
4. On the WSDL Operation screen, check the process operation, and click Next. Use the following image as a guide:



5. Click Next in the WS-Policy Options screen.
6. On the Deploy Policy screen, select the “Default Services” for deployment, and click Finish.
7. Click OK in the Summary window.

8. After the service is registered in the Gateway, review the new artifacts added into the Listeners, Web Service Repository, and Policies containers:



9. Deploy the configuration.

Test the registered service:

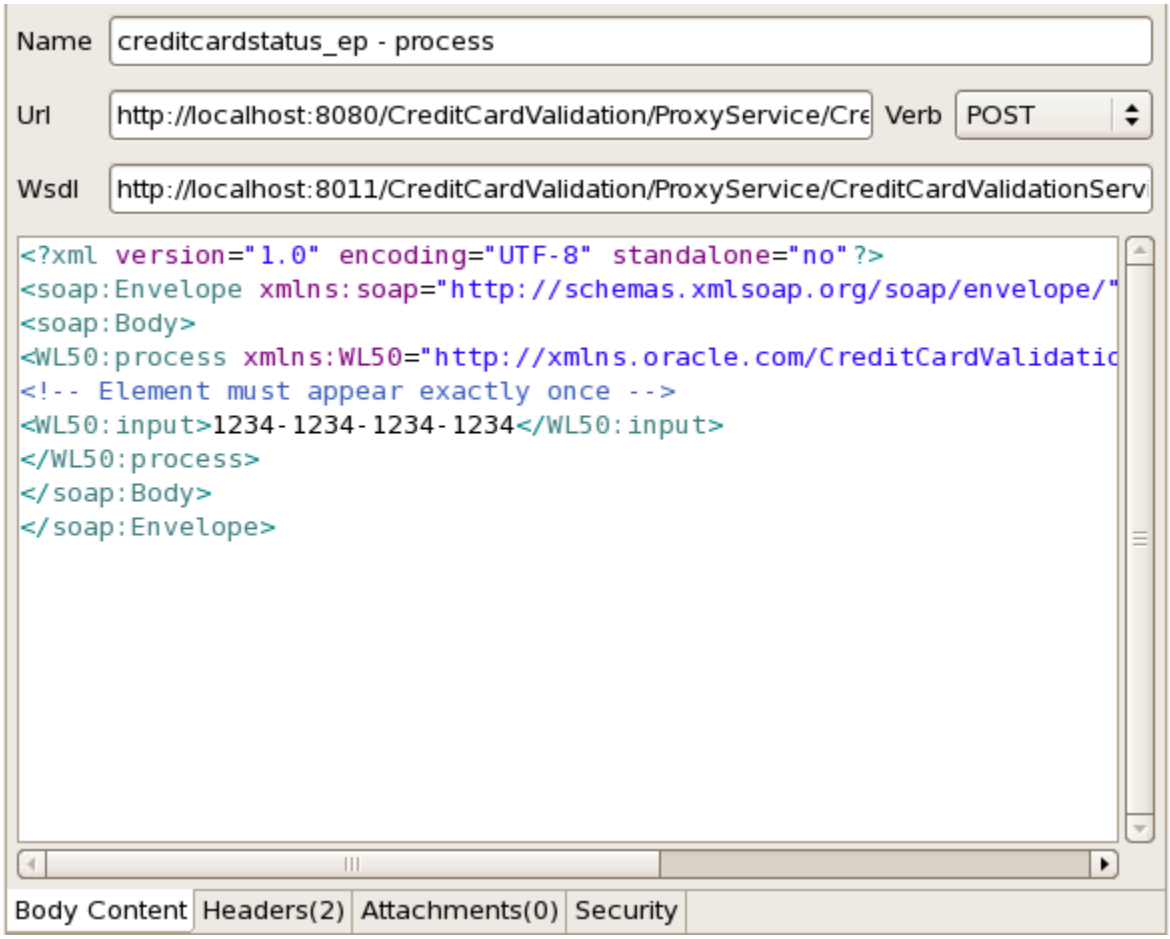
1. Open Service Explorer.
2. Click Import WSDL in the Service Explorer toolbar to launch the Load WSDL wizard.
3. In the Load WSDL screen, select the WSDL URL option, and copy and paste the URL of the proxy service's WSDL file:

   ```
   http://localhost:8011/CreditCardValidation/ProxyService/CreditCardValidationService?wsdl
   ```

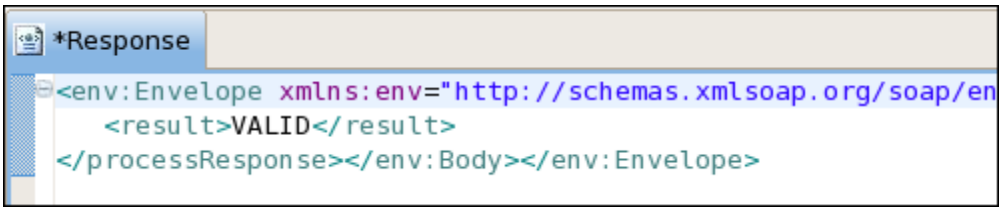
4. Click Next.
5. In the WSDL Operations screen, check the process operation, and click Finish.
6. In the Request tab on the left, you should see that the panel is automatically populated with the SOAP message.
7. Click "Request Settings…" from the drop-down menu.
8. In the Request Settings window, change the URL that serves the request to:

   ```
   http://localhost:8080/CreditCardValidation/ProxyService/CreditCardValidationService
   ```

9. Provide a card number in the request message body. Use the following screenshot as a guide:



10. Send the request. You should see the validation result is returned in the response.



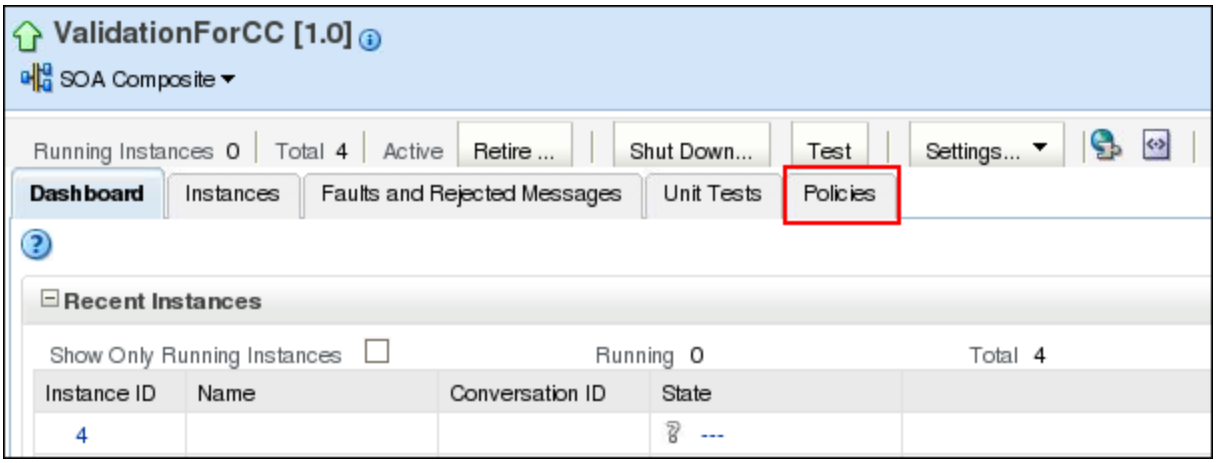
Practice 13-4: Applying an OWSM Security Policy to the Web Service

Overview

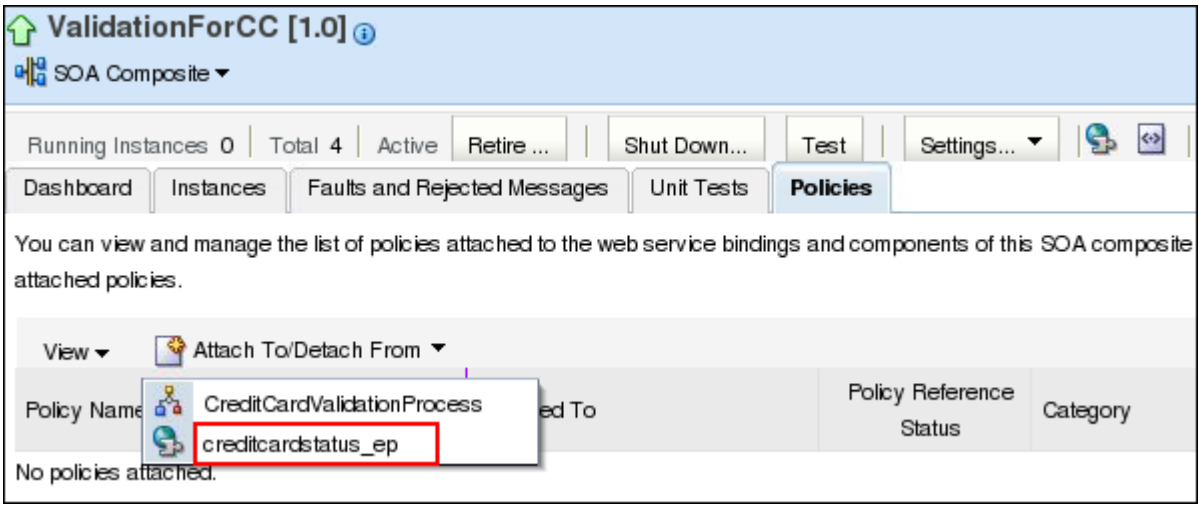
In this practice you attach a WebLogic security policy to the SOA composite application.

Tasks

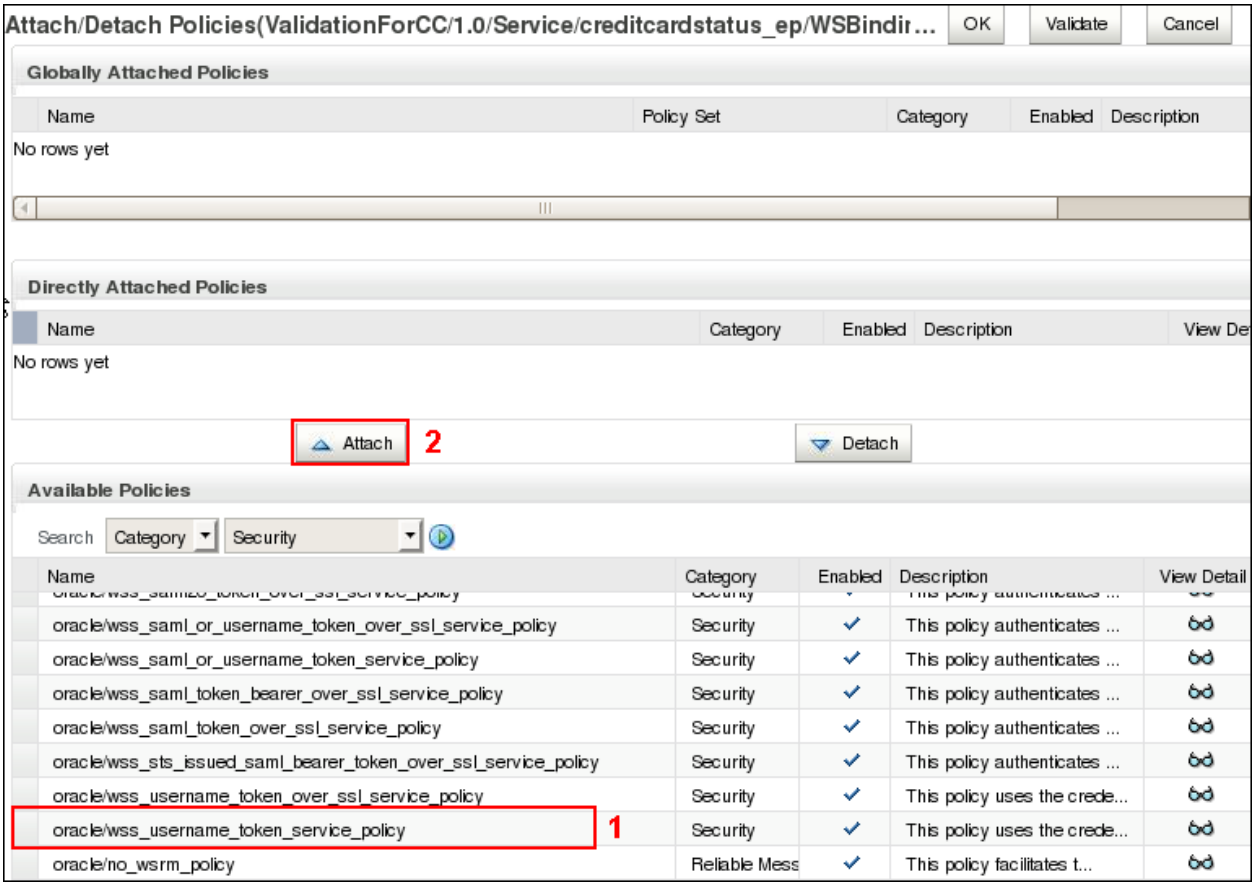
1. In Firefox browser, log in to Enterprise Manager Console at http://localhost:7001/em
2. On the Farm_soa_domain tree, expand SOA > soa-infra > default folder, and click the ValidationForCC link.
3. On the ValidationForCC [1.0] page, click the Policies tab.



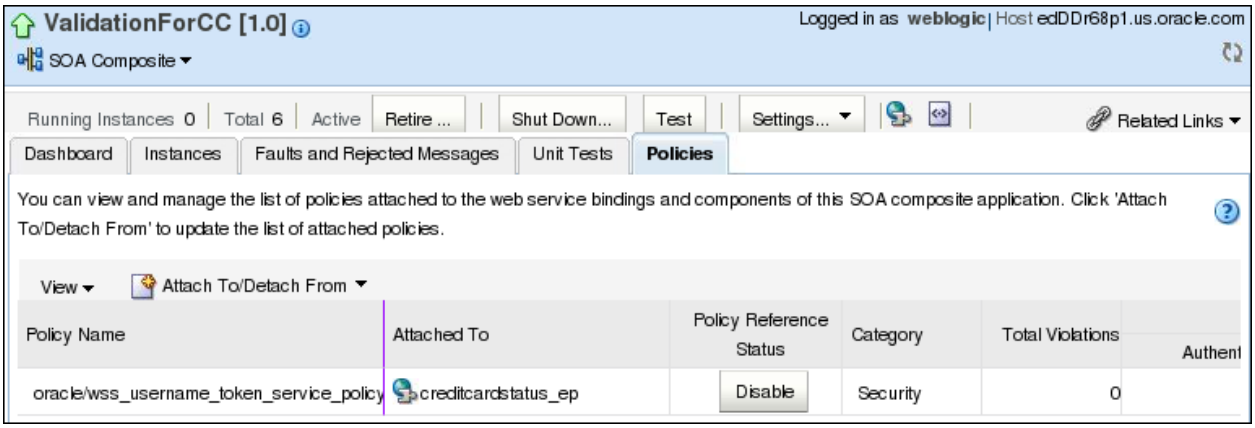
4. On the ValidationForCC [1.0] Policies page, click “Attach To/Detach From” > creditcardstatus_ep.



5. On the “Attach/Detach Policies(CreditCardValidation …)” page, under the Available Policies section, select oracle/wss_username_token_service_policy, and click Attach.



6. You can click Validate (on the top of the page) to validate the attached policy. After you see a “Validation is successful” message, click OK.
7. On the ValidationForCC [1.0] Policies page, you can view the policy and the service endpoint to which the policy is attached in addition to security violation information counters (use the horizontal scroll bar embedded in the page).



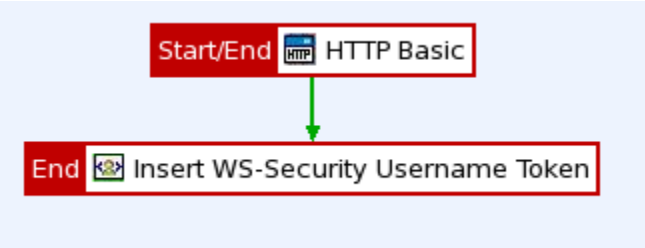
Practice 13-5: Adding a policy to the registered web service in OEG

Overview

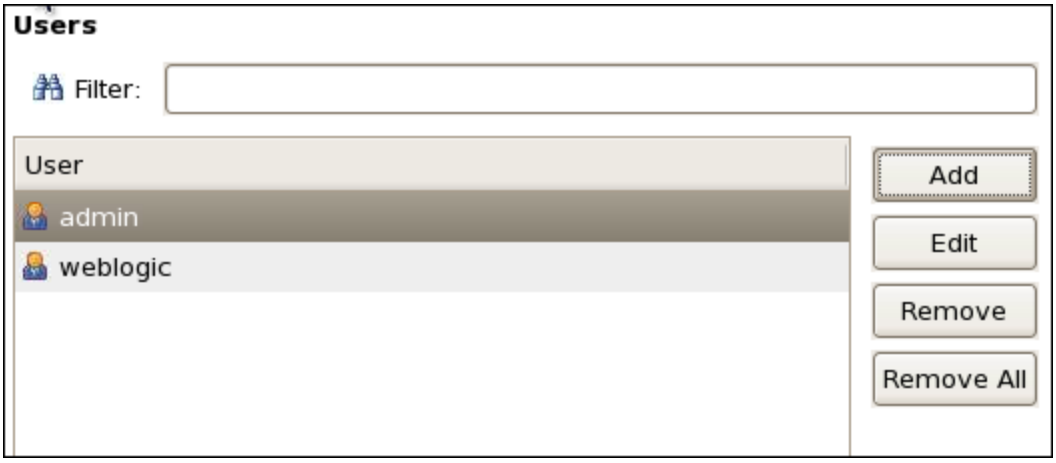
In this practice you add a policy (already created for you) that converts the HTTP basic authentication to WSS Username Token, and apply it to the registered web service in OEG.

Tasks

1. Open Policy Studio and import the `authentication_soa.xml` file located in the labs/Lesson_13 folder.
2. A policy container called Authentication_SOA-OWSM is displayed under the Policies folder. Expand the node, and you should see two policies.
3. Click HTTPBasic-WSSUsernameToken, and view its circuit on the canvas.

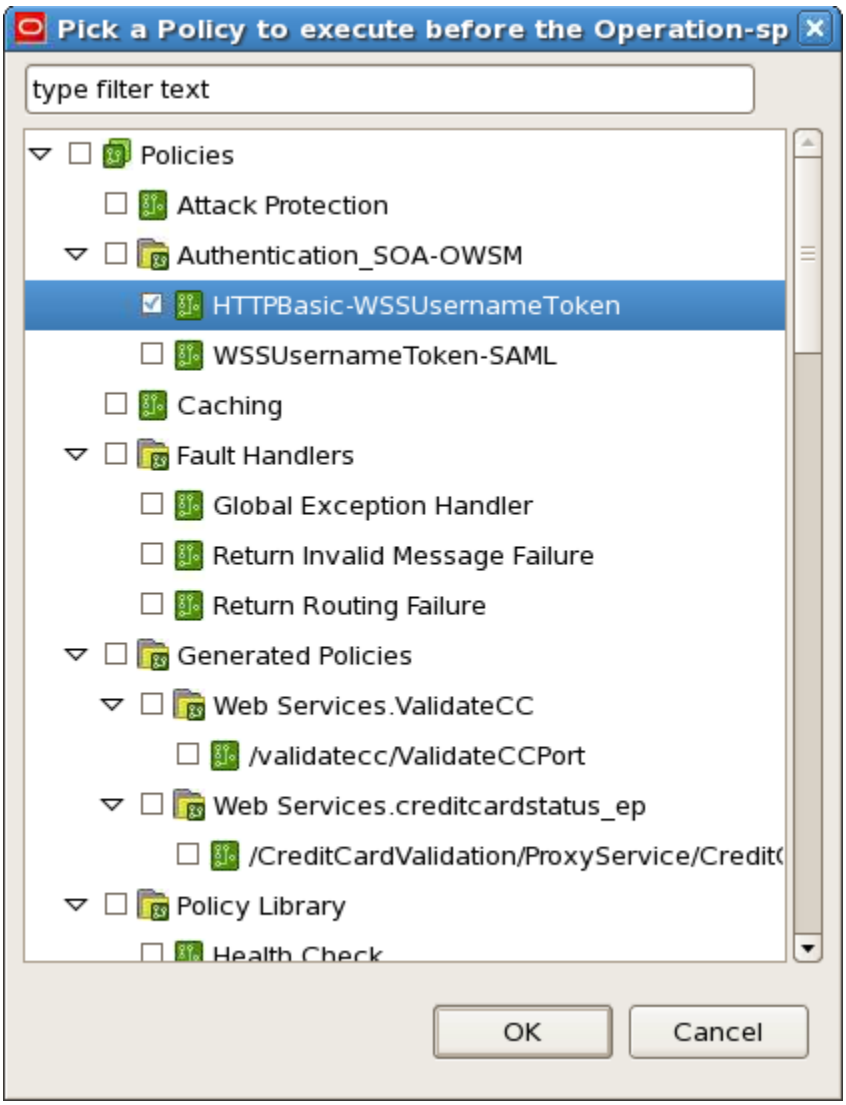


4. In this lab, HTTP Basic authenticates the user against a user profile stored in the Local User Store, so you need to add the user, `weblogic`, into the store. `weblogic` is the admin user of SOA server. To add the user:
   a. Navigate to Users And Groups > Users.
   b. Add a user with User Name: weblogic, and Password: welcome1.



5. Attach the policy to the registered service:
   a. Navigate to the policy generated for the creditcardstatus_ep service under Policies > Generated Policies > Web Services.creditcardstatus_ep
   b. Edit the Service Handler for ‘creditcardstatus_ep’ filter
   c. In the Configure “Service Handler for ‘creditcardstatus_ep’” window, select the Message Interception Points tab.

d. On the Request from client tab, click the ... button, and select the HTTPBasic-WSSUsernameToken policy from the list.



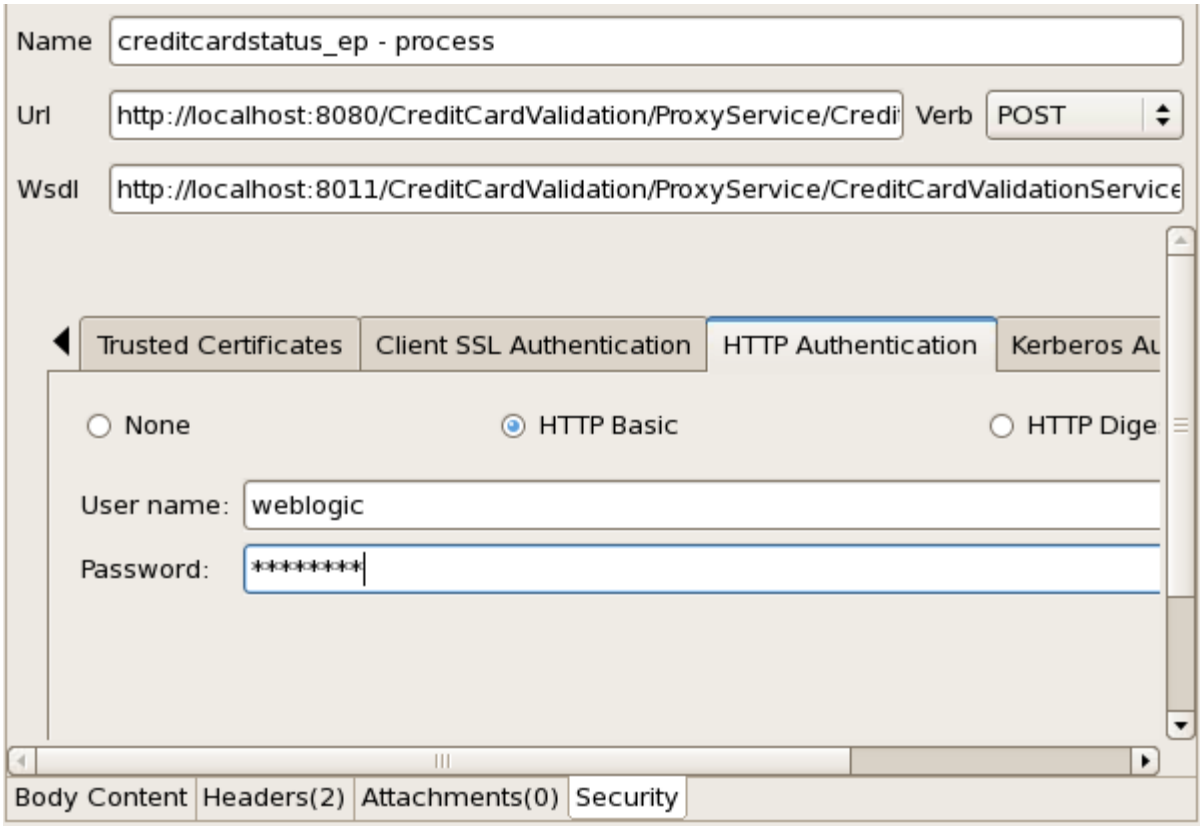
e. Click Finish.

6. Deploy the configuration.

Testing the policy

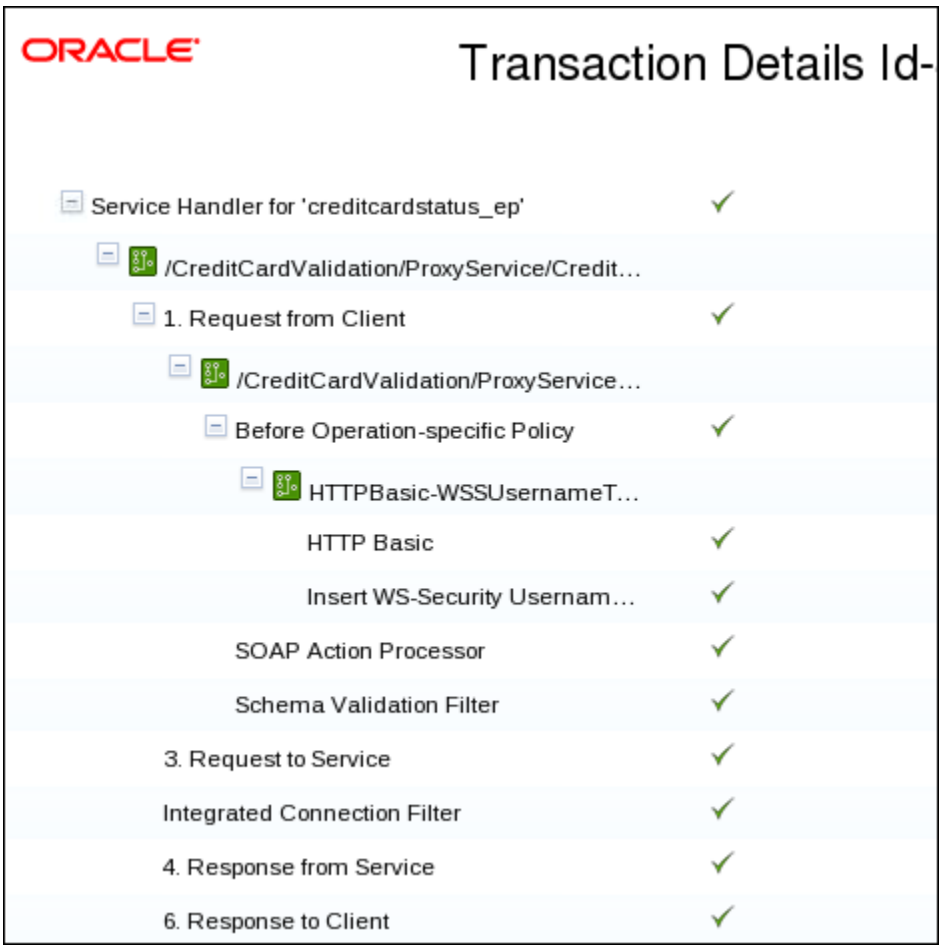
1. Open Service Explorer.
2. Click “Request Settings” and select the creditcardstatus_ep’ request.
3. Select the Security tab at the bottom of the right frame.

4. On the HTTP Authentication tab, click the HTTP Basic option, and enter weblogic/welcome1 as the username and password.



5. Click Run. You should see the response with the validation status same as before without applying the security policy. Test by using another card number, and see the response.

6. You can open a browser, and view the policy flow in the Real Time monitoring and Audit Messages.



Practices for Lesson 14: Integrating with Identity and Access Management

Chapter 14

Practices for Lesson 14: Overview

Practices Overview

There are no practices for this lesson titled “Oracle Enterprise Gateway 11*g*: Security Management and Control for SOA and the Cloud – Integrating with Identity and Access Management”.

General Notes

There are no notes.

Practices for Lesson 15: Securing Services in the Cloud

Chapter 15

Practices for Lesson 15: Overview

Practices Overview

There are no practices for this lesson titled “Oracle Enterprise Gateway 11*g*: Security Management and Control for SOA and the Cloud – Securing Services in the Cloud”.

General Notes

There are no notes.