An Oracle White Paper
June 2011

Oracle Access Manager 11g – Oracle Enterprise Gateway Integration Guide

Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

1 Introduction

1.1. Purpose

This document describes how to configure the OEG Gateway to authenticate and authorize via Oracle Access Manager 11g. This will be demonstrated by the following:

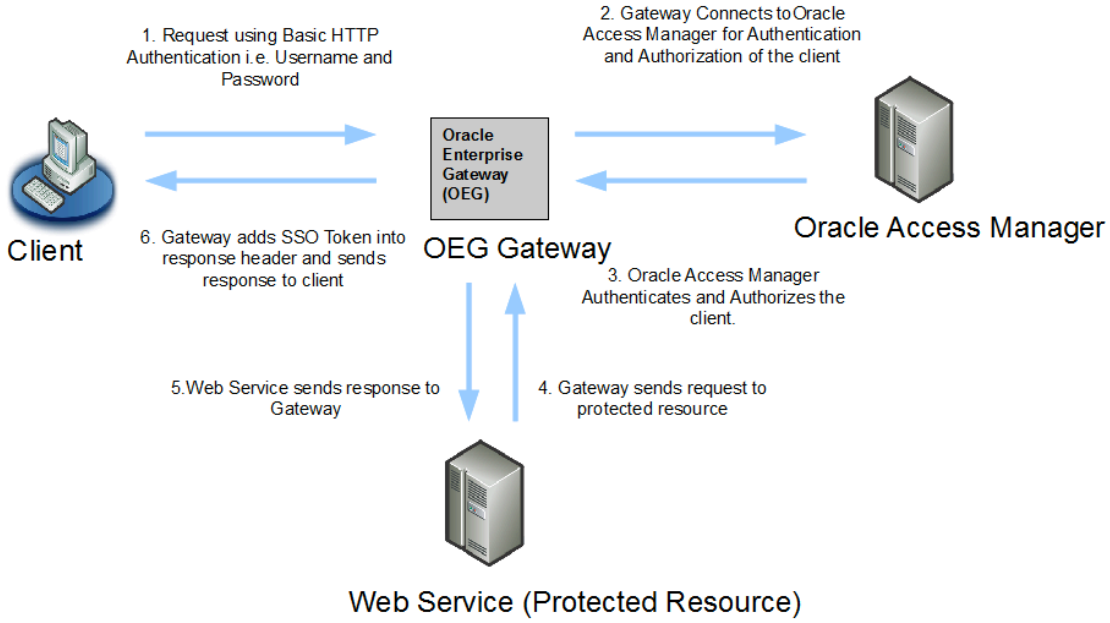
- The OEG Gateway will be configured to authenticate a client via Oracle Access Manager via user name and password contained in an incoming message. These credentials can be found from either HTTP Basic, WS-Security username token, or anywhere inside the message payload.
- Upon successful authentication the Gateway will authorize the user via Oracle Access Manager.
- The Gateway will be also be configured to add an Oracle Access Manager session token into the session HTTP headers or body of the message in order for the message to be validated and authorized without having to authenticate repeatedly via Oracle Access Manager.
- A policy will be created that will look for a Oracle Access Manager session token and make a decision depending on whether an Oracle Access Manager session token is present or not. When no token is found in the message or HTTP headers, the user will be authenticated using HTTP Basic authentication to Oracle Access Manager. If the session token is found in either the message header or body, the user need not be authenticated again and provided the token can be validated, the user will be granted access to the web resource.

Sections in this guide:

- Section 1 explains the general concept of connection OEG Gateway to Oracle Access Manager.
- Section 2 explains the prerequisite step which needs to be carried out on the Gateway before connecting to Oracle Access Manager.
- Section 3 describes the setup of a sample resource that will be protected by Oracle Access Manager.
- Section 4 explains the configuration to connect the Gateway to Oracle Access Manager and the basic policy for authentication and authorization. This section also contains a test of the policy using OEG Service Explorer.

- Section 5 explains the configuration of a policy to check for an existing Oracle Access Manager session token in the message header. This section also contains a test of the policy using OEG Service Explorer.
- Section 6 explains the modification of the policy to check for an existing Oracle Access Manager session token in the message body. This section also contains a test of the policy using OEG Service Explorer.

A graphical representation of the flow of a message through OEG Gateway authenticating and authorizing a user via Oracle Access Manager and finally passing it onto the web service.



This guide applies to OEG software products, from version 6.x upwards. In this guide Oracle Access Manager 11g is used.

1.2. Oracle Access Manager

Oracle Access Manager is a state-of-the-art solution for both centralized identity management and access control, providing an integrated standards-based solution that delivers authentication, web single sign-on, access policy creation and enforcement, user self-registration and self-service, delegated

administration, reporting, and auditing. Oracle Access Manager's unique coupling of access management and identity administration functionality is why it is established as the leading solution for web access management. It excels in complex, heterogeneous enterprise environments and integrates out-of-the-box with all leading directory servers, application servers, web servers, and enterprise applications. Oracle Access Manager is a component of Oracle Fusion Middleware, a well-integrated family of customer-proven software products designed to shine in the most demanding customer environments.

Oracle Access Manager helps enterprises create greater levels of business agility, ensure seamless business partner integration, and enable regulatory compliance. Oracle Access Manager's innovative, integrated architecture uniquely combines identity management and access control services to provide centralized authentication, policy-based authorizations, and auditing with rich identity administration functionality such as delegated administration and workflows. By protecting resources at the point of access and delegating authentication and authorization decisions to a central authority, Oracle Access Manager helps secure web, J2EE, and enterprise applications - such as Oracle PeopleSoft - while reducing cost, complexity, and administrative burdens.

1.3. Setup Used for this Guide:

- OEG Gateway 11.1.1.5.0
- Oracle Access Manager 11g

2 Prerequisites for connecting to Oracle Access Manager

2.1. OEG Gateway as an AccessGate

The OEG Gateway acts as an AccessGate 10g to Oracle Access Manager. AccessGates are Oracle Access Manager clients. They process requests for access to resources within the domain protected by your Access Manager. If a resource is not protected, the AccessGate grants the user free access to the requested resource. If the resource is protected and the user is authorized to provide certain credentials to gain access, the AccessGate attempts to retrieve those user credentials so that the Access Server can validate them. If authentication of the user and authorization for the resource succeed, the AccessGate makes the resource available to the user.

2.2. Install the Access Manager SDK

The Access Manager SDK must be installed on the machine running the OEG Gateway so that the Gateway can act as an AccessGate. You can download the Access Manager SDK from the Oracle Technology Network at the following URL:
http://www.oracle.com/technology
Install the Access Manager SDK which is appropriate to the platform that you are running on.
Important: Remember the location where you have installed the Access Manager SDK as you will need this information later when configuring the OEG Gateway.

On Windows the default installation path is:

C:\Program Files\NetPoint\AccessServerSDK

On Linux it is:

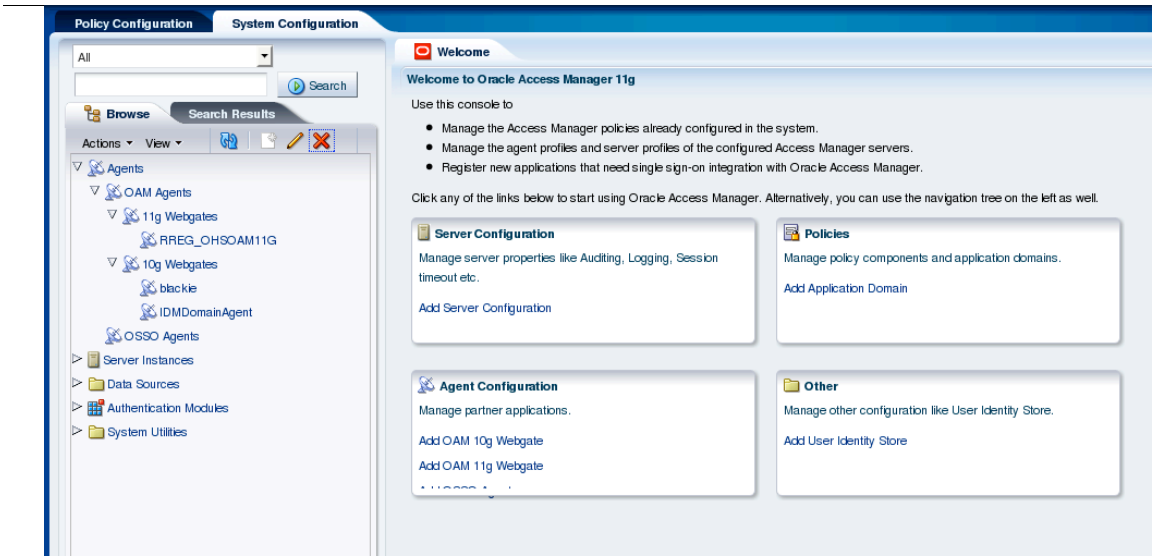
/opt/netpoint/AccessServerSDK

2.2. Configuring the AccessGate

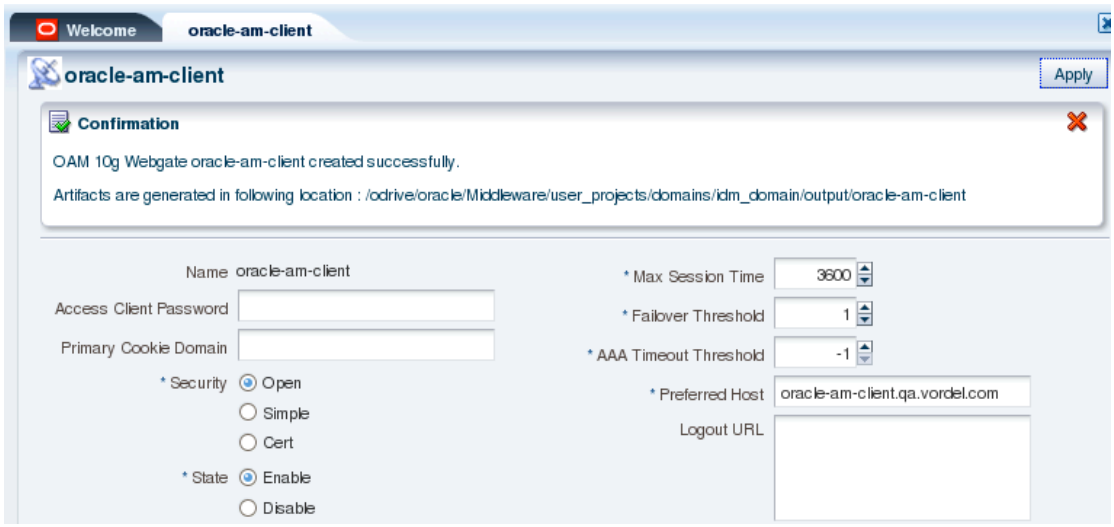
Once the Access Manager SDK has been installed, an AccessGate entry needs to be created on the Access Server.

Creating an AccessGate entry:

- Navigate to the Access Manager Console, switch to "System Configuration" tab and click "Add OAM 10g Webgate".
- Type a convenient name in the AccessGate "Name" field. Choose a name that distinguishes this particular AccessGate from all the others in your system.
- In the "Host Identifier" field, type the DNS name of the machine hosting the server instance on which the AccessGate resides. For example, oracle-am-client.qa.vordel.com
- Type an alphanumeric string for use as a password whenever the AccessGate connects to the Access Server. This value is optional for all transport modes, although the Simple and Cert modes use other passwords not directly related to AccessGate configuration.
- Retype the password to confirm it.
- Click "Apply" button at the top-right corner of the panel to commit the values.



The new AccessGate entry will be created and you'll be notified of the location of the file called "ObAccessClient.xml" which will contains configuration data necessary for the AccessGate to connect to the Access Manager server.

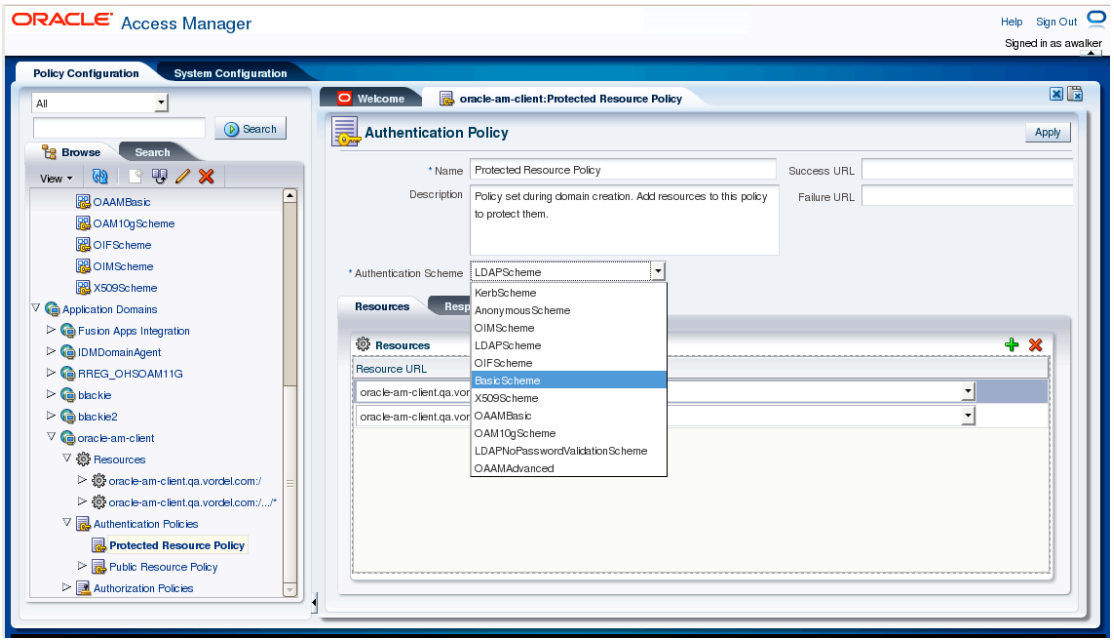


The "ObAccessClient.xml" must be copied to the host where AccessManager SDK is installed, into a following location:

<LOCATION WHERE ACCESS MANAGER SDK INSTALLED>/oblix/lib

Update Authentication Policy to use Basic Scheme:

- Navigate to the Access Manager Console, switch to "Policy Configuration" tab and expand "Application Domains" component.
- In the expanded subtree locate entry with the same name as the newly created AccessGate entry
- Expand "Authentication Policies" node, select "Protected Resource" Policy, and click on the "Edit" icon (located on the top of the tree)
- Change "Authentication Scheme" to "BasicScheme" and click "Apply" button in the top right corner.
- You might need to restart Access Manager for changes to take effect.



You have now successfully set up the AccessGate configuration so that the OEG Gateway can perform authentication and authorization services on the Oracle Access Server.

3 The Protected Resource: Sample Service

Details of the Sample Service:
OEG Gateway ships with an axis sample web service. For the purpose of this document it is this service that is being protected by Oracle Access Manager and message will be routed to this sample service to simulate a more realistic environment for testing purposes.
Details of Sample Service:
OEG Gateway ships with an axis sample service. For the purpose of this demonstration it is this service that is protected by Access Manager.
Details of Sample Service:

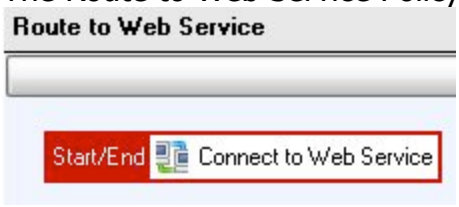
- The axis service can be started by running the **axissimpleserver.bat** file located in the **/Gateway_Install_Dir/win32/bin** for Windows or **axissimpleserver.sh** located in the **/Gateway_Install_Dir/posix/bin** for Unix/Solaris
- For the purpose of this guide the axis service will be run on the local machine on **port 7070** which is the default port for this service.
- The URL for the resource is: **/axis/services/urn:xmltoday-delayed-quotes**
- The full URL for the service will be: **http://host_ip:7070/axis/services/urn:xmltoday-delayed-quotes**
- The WSDL for the available services can be retrieved from: **http://host_ip:7070/axis/services/urn:xmltoday-delayed-quotes?wsdl**
- The Service home page is**: http://host_ip:7070/**
-

Creating the Routing Policy:

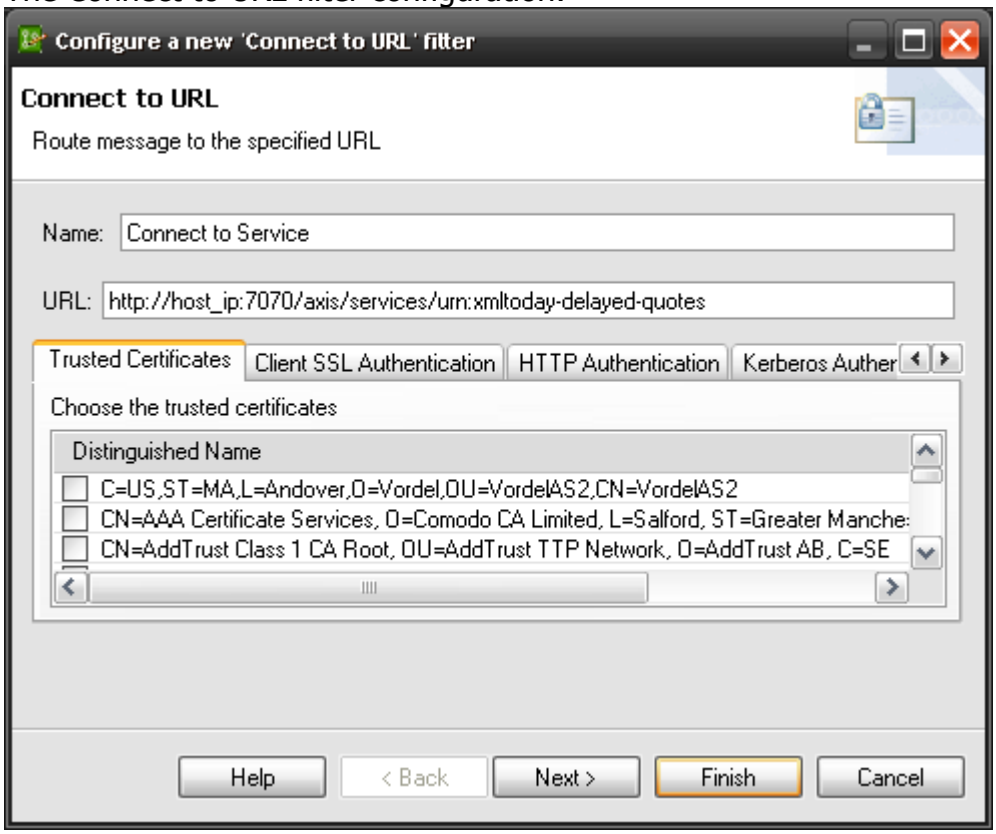
1. Start Policy Studio by running **"policystudio.exe"** (Windows) or **"policystudio.sh"** (Unix/Solaris) from the Policy Studio root directory.
2. Click the URL for the Gateway or Policy Director where Gateways are managed via Policy Director.
3. Click on the OEG Gateway process listed to open the configuration window in a new tab.
4. Click on the **"Policies"** module and right click on **"Policies"** and select **"Add Policy".**
5. Drag a **"Connect to URL"** filter from the **"Routing"** filter category.

6. Rename the Name of the filter to: **Route to Web Service**
7. Enter URL: **http://host_ip:7070/axis/services/urn:xmltoday-delayed-quotes**
8. Click on **"Finish".**

The Route to Web Service Policy:



The Connect to URL filter configuration:



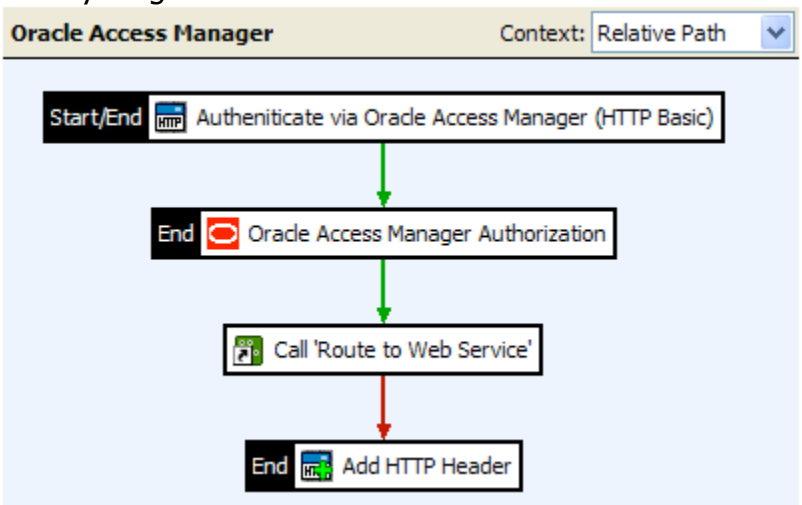
4 Configuring the OEG Gateway

Create a Policy to Authenticate and Authorize a User via Oracle Access Manager
The policy will authenticate and authorize an user via Oracle Access Manager. An Oracle Access Manager session token will also be added to the HTTP Header of the message before it is passed back the client.

The flow of the policy:

1. User will present a username and password combination to the Gateway using HTTP Basic authentication. The user's credentials will be passed to Oracle Access Manager, which will authenticate the user.
2. The user will be authorized via Oracle Access Manager for a particular resource in this case **/axis/services/urn:xmltoday-delayed-quotes.**
3. If the above steps are successful the message will be routed to the web service via the policy shortcut that will call the Route to Web Service policy.

The Policy diagram:



4.1 Creating the policy:

Step 1: Configure an Oracle Authentication Repository

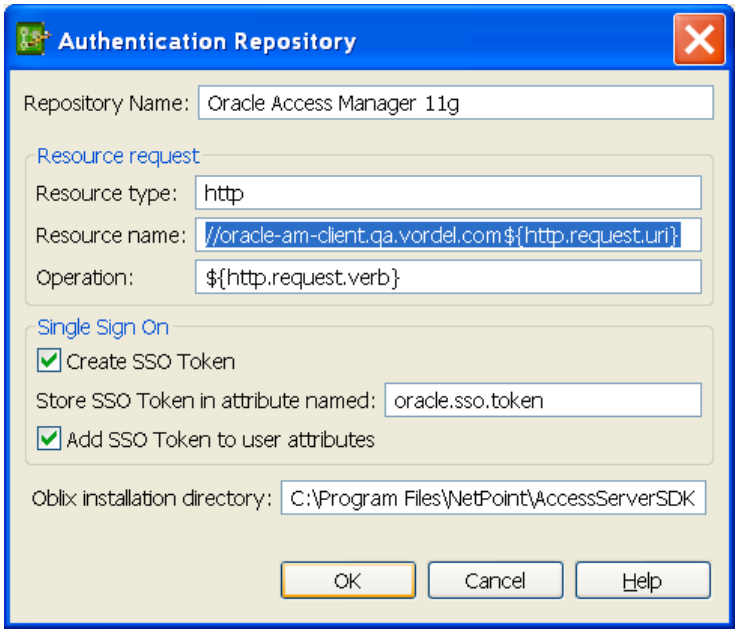
Step 2: Create a HTTP Basic Authentication filter to authenticate the user against Oracle Access Manager
Step 3: Configure an Oracle Access Manager Authorization Filter
Step 4: Configure a Policy Shortcut filter and link it to Route to the Route to Web Service Policy
Step 5: Add the SSO Session Token to the HTTP Header of the response.
Step 6: Create relative path to policy
Step 7: Test the Policy

Step 1: Configure Oracle Access Manager Authentication Repository:

- Start Policy Studio by running "policystudio.exe" (Windows) or "policystudio.sh" (Unix/Solaris) from the Policy Studio root directory.
- Click on the URL for the Gateway or Policy Director if the Gateway/s is managed via Policy Director.
- Click on the OEG Gateway process listed to open the configuration window in a new tab.
- Click on the "External Connections" module.
- Expand "Authentication Repository Profiles" and right click on "Oracle Access Manager Repository" then select "Add a new Repository".
- Provide a name for the Repository. "Oracle Access Manager 11g" is used for this guide.
- For "Resource request" section configure the details of the resource which is to be secured:
  1. "Resource type" represents the type of resource being requested. The default value is HTTP.
  2. "Resource name" which is the name of the resource. Set it to following "**//<host-identifier>${http.request.uri}**" where **<host-identifier>** is the "Host Identifier" value from Section 2.2. The "${http.request.uri}" means that the value of the path of the incoming HTTP message. So in this example the resource name will be "//oracle-am-client.qa.vordel.com/sampleService".
  3. "Operation", which is the type of operation to be performed against the resource. When the resource type is HTTP, the possible operations are GET and POST.
  4. Under the "Single Sign On " section a "Single Sign On" (SSO) token can be created and stored in a Vordel message attribute, so that is can be added later to the response that the client receives. Note that it is also possible to add this SSO token to the user attributes so that

it can be easily inserted into a SAML attribute assertion. Check the box for creating a SSO token and leave all other fields as default.

- Enter the location of where the Access Manager SDK has been installed in the "Oblix installation directory" i.e. the default
  on Windows is:
  C:\Program Files\NetPoint\AccessServerSDK
  On Linux systems it is:
  /opt/netpoint/AccessServerSDK



Step 2: Create a HTTP Basic Authentication filter to authenticate the user against Oracle Access Manager

1. Add a new policy by clicking on the "Policies" module then right click on "Policies" and select "Add Policy". Name the Policy "Oracle Access Manager".
2. Add a "HTTP Basic" filter located in the "Authentication" filter category and configure it as follows:

HTTP Basic Filter Configuration:

- Name: of the filter can be left default or changed to any descriptive name.

- Realm: Populated automatically by the value specified in System Settings in the Gateway.
- Credential Format: select "User Name" from the drop down list.
- Repository Name: The Oracle Access Manager repository must be selected here as configured above. Select "Oracle Access Manager 11g" from the drop down list.
- Click on "Finish".

Step 3: Configure a Oracle Access Manager Authorization Filter

- Add an "Authorization" filter from the "Oracle Access Manager" filter category.
- The rest of the settings can be left as default, except for the "Resource Name" which should be set to the same value as in Step 1. Click "Finish".
- Connect the "Oracle Access Manager Authorization" filter to the "HTTP Basic" filter with a success path.

Step 4: Configure a Policy Shortcut to call point to Route to Web

- Drag a "Policy Shortcut" filter from the "Utility" filter category.
- Choose the "Route to Web Service" policy that will route the message to the sample web service.
- Click on "Finish".
- Connect the "Policy Shortcut" filter to the "Oracle Access Manager Authorization" filter with a success path.

Step 5: Add the session to the response

- Add an "Add HTTP Header" filter from the "Conversion" filter category.
- For the "HTTP Header Name" field value enter "ssosession", this will be the name of the HTTP header that will be added to the response.
- Give the "HTTP Header Value" field the value "${oracle.sso.token}", this result in the SSO token generated by Oracle Access Manager been placed in the header "ssosession".
- Click on "Finish".
- Connect the 'Add HTTP Header' filter to 'Policy Shortcut' filter with a success path.

- Deploy the new configuration to the server by pressing "F6" on the keyboard or by clicking on "Settings" at the top and select "Deploy".

Step 6: Configure Relative Path to Policy

- Click on the "Services" module in Policy Studio.
- Expand "Processes", "OEG Gateway" and right click on the "Default Services".
- Select "Add Relative Path" and enter: /axis/services/urn:xmltoday-delayed-quotes
- Select the "Oracle Access Manager" Policy from the policy list.
- Deploy the new configuration to the server by pressing "F6" on the keyboard or by clicking on "Settings" at the top and select "Deploy".

Step 7: Test Policy with OEG Service Explorer

OEG Service Explorer is a free stress and security tool for web services developed by Vordel. It will be used as the client for testing.
OEG Service Explorer can generate a request via the available web service:

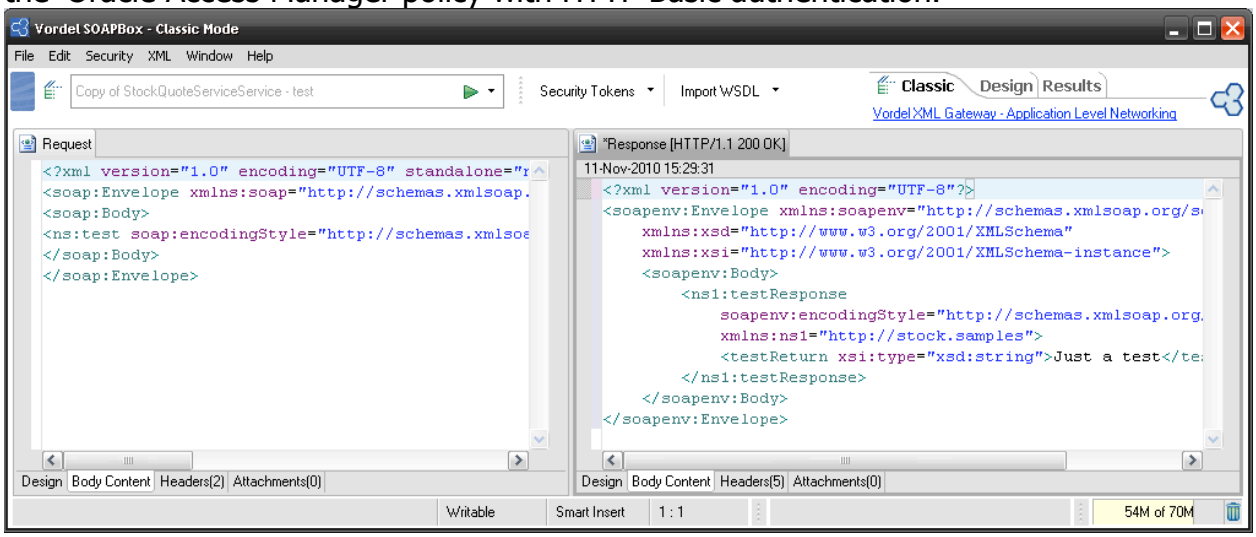
- Start OEG Service Explorer by running "oegserviceexplorer.exe" (win32) or "oegserviceexplorer .sh" (UNIX) located in the OEG Service Explorer root directory.
- Click on the "Import WSDL" button on the top tool bar.
- Select the "WSDL URL" option and enter the URL of the Axis Service WSDL.
- If running the Axis Service locally as described in section 3 the URL is:
- http://localhost:7070/axis/services/urn:xmltoday-delayed-quotes?wsdl
- Click on "OK".
- A request will be automatically generated.
- Change the URL field to the Gateway IP address and port as follows:
- http://localhost:8080/axis/services/urn:xmltoday-delayed-quotes
- Click on the "Security" tab then on the "HTTP Authentication" tab and enter the username and password of a user configured in Oracle Access Manager.
- Then click on the "Run" button to send the request.

Alternatively the following request can be used:

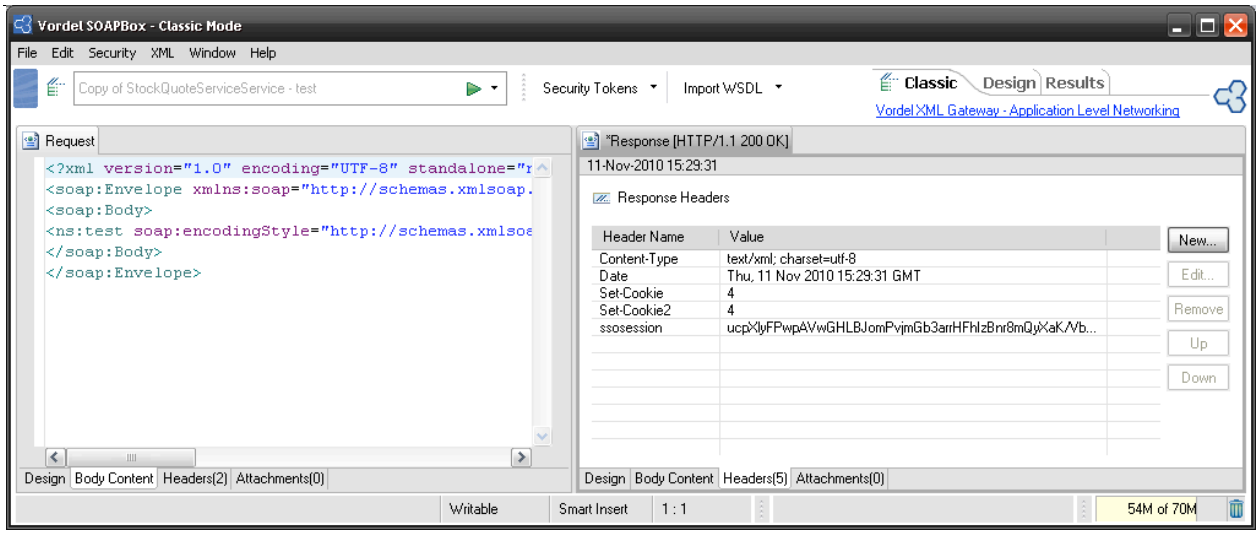
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
<ns:test
soap:encodingStyle="http://schemas.xmlsoap.org/soap/encodi
ng/" xmlns:ns="http://stock.samples"/>
</soap:Body>
</soap:Envelope>
```

The result in OEG Service Explorer after the message has been sent through the 'Oracle Access Manager policy with HTTP Basic authentication.



Note the "ssosession" header in the response, this is the SSO token generated by Oracle Access Manager:



Section 4 Summary:

The policy demonstrates:

- The message will pass through the HTTP Basic filter to authenticate to Oracle Access Manager.
- Once the user has been successfully authenticated the message passes through the authorization filter where the user session is authorized for access to the protected web service.
- The message then gets routed to the web service.
- The message response passes through an Add HTTP Header filter to add a valid session token to the header of the message and then gets passed back to the client.

5 Retrieving a SSO Token from the Message Header

In the previous section the policy has been configured to authenticate, authorize and add a session token to the HTTP Header of the message.

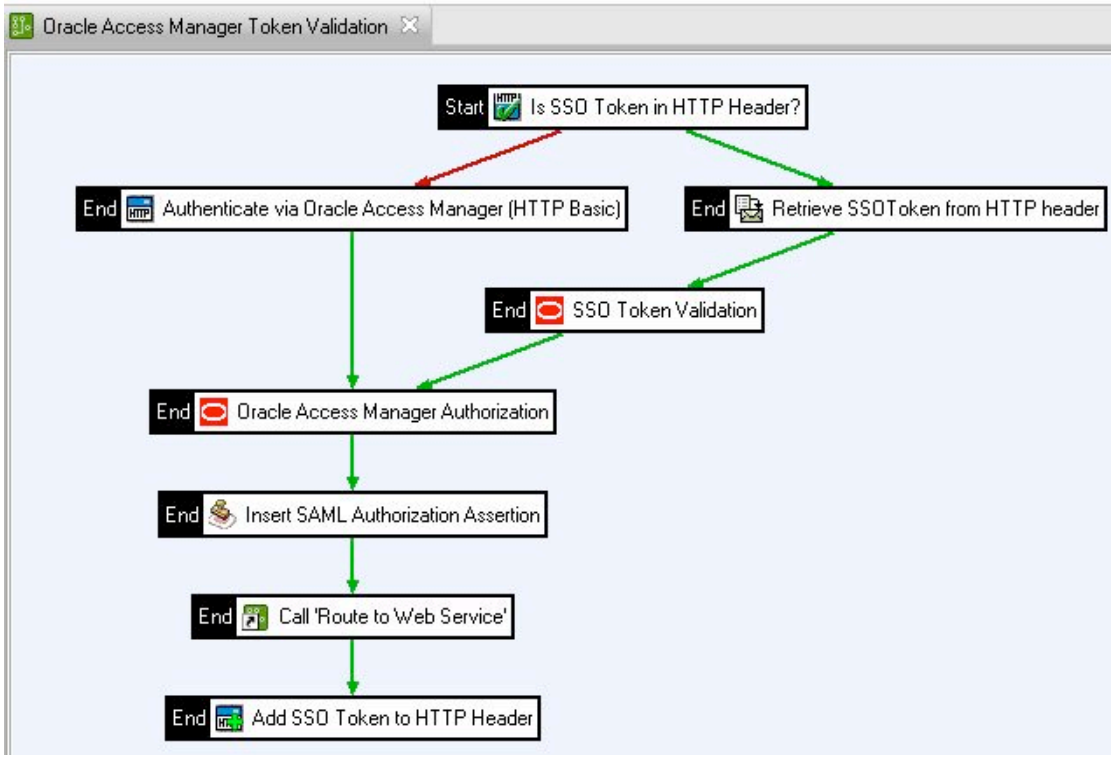
In this section a policy will be created that will look for an existing SSO session token in the message. When a session token is already present in the message (header or body), the request does not need to be authenticated again.

In a real world deployment scenario, it is not practical to have to authenticate and authorize each user for every request they send. By inserting a SSO session token into a message, the token in the message is validated instead of re-authenticating the user for every following request. A SAML Authorization Assertion will also be inserted for consumption by a downstream web service to show that the Gateway has authorized the message.

The flow of the policy will be as follows:

1. Check the request's HTTP headers for the presence of a SSO token.
2. When the token is present, the "HTTP Header Attribute" filter is run to retrieve the value of the token from the "oracle.sso.token" Vordel message attribute. Otherwise, if the token can not be found, the request will be passed to the "HTTP Authentication" filter where the user will be authenticated via Oracle Access Manager using username and password credentials.
3. When the attribute containing the token has been retrieved successfully the session will be validated using the Oracle Access Manager "Session Validation" filter to check if the user token is still valid.
4. When the token has been successfully validated then the user will be authorized using the Oracle Access Manager "Authorization" filter.
5. A SAML Authorization Assertion will be inserted into the message for consumption by downstream web services.
6. The message will then be routed to the web service.
7. If the Message did not have the SSO token in the header it will be passed through the "HTTP Basic" filter the user will be authenticated based on user credentials.
8. The user will then be authorized using the Oracle Access Manager "Authorization" filter.
9. A SAML Authorization Assertion will be inserted into the message for consumption by downstream web services.
10. The message will then be passed to the Route to Web Service policy shortcut that will route the message to the web service. The response message will pass through an "Add HTTP Header" filter that will add a HTTP Header to the message response containing a SSO token.

The flow of the Oracle Access Manager Token Validation policy:



The first step will be to create another Route to Web Service Policy that will also add the session token into the message.

5.1 Creating the Policy:

Success Path summary:
Step 1: Configure a "Validate HTTP Headers" filter to validate that an existing token exists in the HTTP Header of the message.
Step 2: Configure a "Retrieve from Attribute" filter that will retrieve the existing token attribute from the HTTP Header of the message.
Step 3: Configure an "Oracle Access Manager Session Validation" filter that check for the validity of the SSO token.
Step 4: Configure an **"**Oracle Access Manager Authorization" filter that will authorize the client for the protected resource.
Step 5: Configure an "Insert SAML Authorization Assertion" filter that will add a SAML Authorization Assertion to the message for consumption by downstream service.

Step 6: Configure a "Policy Shortcut" filter and link it to the Route to Web Service policy.

Failure Path summary:
Step 1: Create a HTTP Basic Authentication filter to authenticate the user against Oracle Access Manager
Step 2: Configure an **"**Oracle Access Manager Authorization" filter that will authorize the client for the protected resource.
Step 3: Configure an "Insert SAML Authorization Assertion" filter that will add a SAML Authorization Assertion to the message for consumption by downstream service.
Step 4: Configure a "Policy Shortcut" filter and link it to the Route to Web Service.
Step 5: Change relative path of service to point to the Oracle Access Manager Token Validation Policy.
Step 6: Test the policy.

Configuring the success path filters:

Step 1: Configure a "Validate HTTP Headers" filter to validate that an existing token exists in the HTTP Header of the message.

- Add a new policy by clicking on the "Policies" module then right click on "Policies" and select "Add Policy". Name the Policy "Oracle Access Manager Token Validation".
- Add a "Validate HTTP Headers" filter from the "Content Filtering" filter category in the filter palette.
- The filter name: Is SSO Token in HTTP Header?
- Under the "Enter a Regular Expression" section click on the "Add" button.
- A Window titled "Configure Regular Expression" will open. Enter "ssosession" in the name field and select the "Required" radio button.
- - Click on "OK" then right click on the filter and select "Set as Start".

Step 2: Configure a "Retrieve from HTTP Header" filter that will retrieve the existing token attribute from the HTTP Header of the message.

- Add a "Retrieve from HTTP Header" filter from the "Attributes" filter category.

- Double click on the filter to edit it. In the "HTTP Header Name" field enter "ssosession".
- In the "Attribute ID" field enter "oracle.sso.token" then click on "Finish".
- Connect the "Validate HTTP Header" filter to the "Retrieve SSO Token from HTTP Header" filter with a success path.

Step 3: Configure an "SSO Token Validation" filter that checks for the validity of the SSO token.

- Add a "SSO Token Validation" filter located in the "Oracle Access Manager" filter category. In the "Attribute containing SSO token id" field enter the name of the attribute that was configured in the "Retrieve from HTTP Header" filter namely, "oracle.sso.token".
- Enter the location of where the Access Manager SDK has been installed in the "Oblix installation directory" i.e. the default on Windows is:
  C:\Program Files\NetPoint\AccessServerSDK
  On Linux systems it is:
  /opt/netpoint/AccessServerSDK
- Click on "Finish".
- Connect the "Retrieve from HTTP Header" filter to the "Session Validation" filter with a success path.

Step 4: Configure an "Oracle Access Manager Authorization" filter that will authorize the client for the protected resource.

- Add an "Authorization" filter from the "Oracle Access Manager" filter category.
- Configure settings with the same values as configured in the previous policy. Click on "Finish".
- Connect the "SSO Token Validation" filter to the "Authorization" filter with a success path.

Step 5: Configure an "Insert SAML Authorization Assertion" filter that will add a SAML Authorization Assertion to the message for consumption by downstream service.

- Add an "Insert SAML Authorization Assertion" filter located in the "Authorization" filter category.
- Configure the filter as follows:
- Expiry Date: Set to any desired value.
- SOAP Actor/Role: Choose "Current Actor/Role Only" from the drop down list.

- SAML Version: Select the version of SAML required. Options are 1.0, 1.1 or 1.2.
- For "Issuer Name" select the desired issuer from the drop down field.
- For Resource enter: /axis/services/urn:xmltoday-delayed-quotes
- For Action enter read, write or execute. All can be entered separated by commas. For more information click on the "Help" button on the filter.
- Click on the "Confirmation Method" tab and select the desired confirmation method from the list. Please click on the "Help" button for more information on the different options.
- The "Advanced" tab contains more options in regards to layout, using Security Token Reference etc.
- Click on "Finish" once the filter is configured as desired.

Step 6: Configure a "Policy Shortcut" filter and link it to the Route to Web Service policy.

- Add a "Policy Shortcut" filter from the "Utility" filter category in the filter palette.
- Choose the "Route to Web Service" policy that will route the message to the sample web service.
- Click on "Finish".
- Connect the "Policy Shortcut" filter to the "Insert SAML Authentication Assertion" filter with a success path.

Step 7: Add the SSO Session Token to the HTTP Header of the Response.

- Add an "Add HTTP Header" filter from the "Conversion" filter category.
- Give the "HTTP Header Name" field the value "ssosession", this will be the name of the HTTP header that will be added to the response.
- Give the "HTTP Header Value" field the value "${oracle.sso.token}", this results in the SSO token generated by Oracle Access Manager being placed in the header "ssosession".
- Click on "Finish".
- Connect the "Add HTTP Header" filter to "Policy Shortcut" filter with a success path.

Configuring the failure path filters:

Starting back at the top of the policy the failure path will now be configured.

Step 1: Create a HTTP Basic Authentication filter to authenticate the user against Oracle Access Manager

- The 'HTTP Basic' filter used in the first 'Oracle Access Manager Policy' can be used for this policy so simply copy and paste the 'HTTP Basic' filter into the policy canvas.
- Connect the 'Validate HTTP Headers' filter to the 'HTTP Basic' filter with a failure path.

Step 2: Configure an 'Oracle Access Manager Authorization' filter that will authorize the client for the protected resource.

- Connect the success path of the "Validate HTTP Headers" to the existing "Oracle Access Manager Authorization" filter.
- Deploy the new configuration to the server by pressing "F6" on the keyboard or by clicking on "Settings" at the top and select "Deploy".

Step 3: Change relative path to "Oracle Access Manager Token Validation" policy

- Change the /axis/services/urn:xmltoday-delayed-quotes relative path in the Gateway to point to "Oracle Access Manager Token Validation" policy and click on "OK".
- Deploy the new configuration to the server by pressing "F6" on the keyboard or by clicking on "Settings" at the top and select "Deploy".

5.2 Testing the policy with OEG Service Explorer

To test the modified policy the same procedure will be used as before. However, a message will also be sent through that already contains the SSO token in a header called "ssosession".
OEG Service Explorer can generate a request via the available web service:

- Start OEG Service Explorer by running "oegserviceexplorer.exe" (win32) or "oegserviceexplorer.sh" (UNIX) located in the OEG Service Explorer root directory.

- Click on the "Import WSDL" button on the top tool bar.
- Select the "WSDL URL" option and enter the URL of the Axis Service WSDL.
- If running the Axis Service locally as described in section 3 the URL is:
- http://localhost:7070/axis/services/urn:xmltoday-delayed-quotes?wsdl
- Click on "OK".
- A request will be automatically generated.
- Change the URL field to the Gateway IP address and port as follows:
- http://localhost:8080/axis/services/urn:xmltoday-delayed-quotes
- Click on the "Security" tab then on the "HTTP Authentication" tab and enter the username and password of a user configured in Oracle Access Manager.
- Then click on the "Run" button to send the request.

Alternatively the following request can be used:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soap:Body>
<ns:test soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns="http://stock.samples"/>
</soap:Body>
</soap:Envelope>
```

The result in OEG Service Explorer after the message has been sent through the 'Validating Message Header' policy with HTTP Basic authentication. The Oracle SSO session token can be seen in the HTTP Header of the response below.

The next message that will be sent through will be sent with the Oracle SSO token inserted into the header.

- Copy the value from the "ssosession" section.
- Enter the URL for the Gateway and resource path. In this case: http://GATEWAY_HOST:8080/axis/services/urn:xmltoday-delayed-quotes (where "GATEWAY_HOST" points to the IP address or hostname of the machine running the XML Gateway.)

- Copy the test message into the SOAP Request window.
- Click on the "Headers" tab in the Request window and click on "Add".
- In the "Name" field enter "ssosession" and in the "Value" field paste the string that was copied above. This is actually the same session value that was inserted and viewed in the "Headers" tab of the returned message.
- Click on "Settings" and "Connection Settings" above the "Send Request" button. Click on the "HTTP Basic" and select "None". Click on "Finish" to send the message. The message will now be sent through without the user credentials but with an Oracle SSO session token inserted into the HTTP header of the message.
- Click on "Finish" to send the message.

The message will be sent successfully but this time the client sending the message was validated using the SSO token.

Section 5 Summary:

In this section the policy demonstrates how the message can follow two different paths depending on whether a SSO token is present in the header or not.

Success Path: Token is present in the http header of message

- Message will be checked for the presence of a valid token.
- Token will be retrieved from header of message.
- The Oracle Access Manager SSO token will be validated.
- The user will then be authorized.
- The message will pass through an "Insert SAML Authorization Assertion" filter for consumption by a downstream web service.
- The message will then be routed to the sample web service.

Failure Path: Token is not present in the http header of message

- Message will be checked for the presence of a valid token.
- If the token is not present then the message will follow the failure path to the HTTP Basic filter where the user will be authenticated if a valid username and password are present.
- When the user credentials have been successfully authenticated the message will pass to the Authorization filter where the user will be authorized.

- The message will pass through an "Insert SAML Authorization Assertion" filter for consumption by a downstream web service.
- The message will then be routed to the sample web service.
- The SSO token will be added to the HTTP header of the response.

6 Conclusion

This document demonstrated how to configure the OEG Gateway to authenticate and authorize users against Oracle Access Manager. It also demonstrated how the session token can be used to authorize the user without having the client to resend it's credentials again which demonstrated the single sign on capability provided by integrating OEG Gateway with Oracle Access Manager.
This configuration can be part of a larger policy, including features such as XML threat detection and conditional routing, features which are out of the scope of this document but are covered in other documents which can be obtained from Oracle at http://www.oracle.com

Oracle Enterprise Gateway
May 2011
Author:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Oracle is committed to developing practices and products that help protect the environment

SOFTWARE. HARDWARE. COMPLETE.