An Oracle White Paper
May 2011

Oracle 10g XE Database-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.

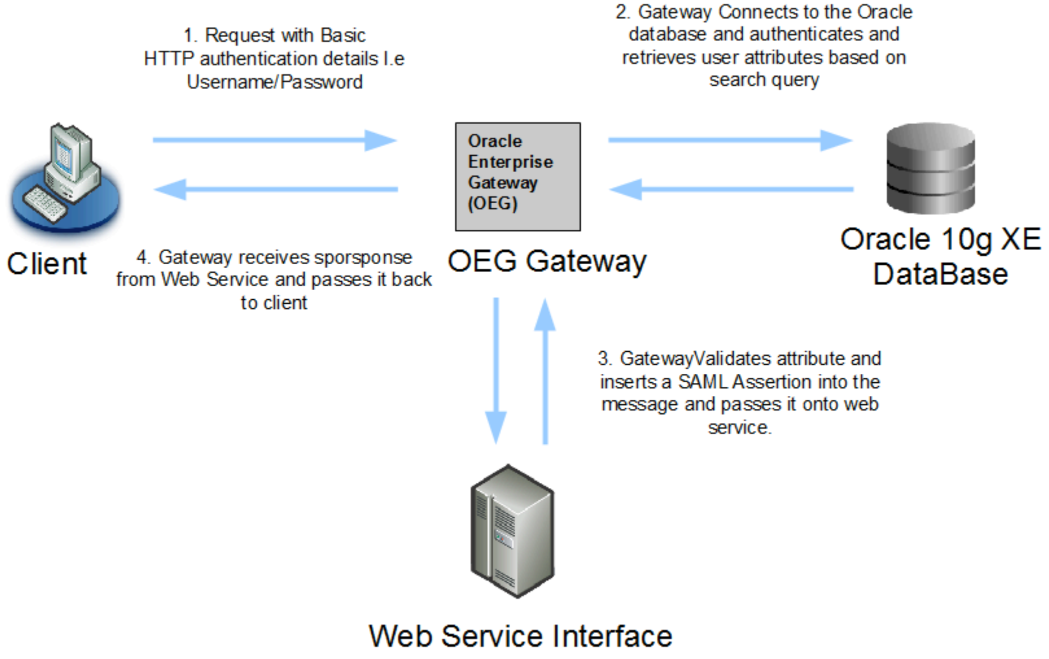
1. Introduction

1.1. Purpose

This document describes how to configure the XML Gateway to authenticate and authorize a user to access a Web Service based on the data stored in an Oracle 10g database.

1. The XML Gateway will be configured to authenticate a user located in the Oracle 10g Express Edition database.
2. Upon successful authentication the XML Gateway will be configured to extract attributes/roles belonging to this user from an Oracle 10 g Express Edition database.
3. The user will be authorized based on the attributes/roles retrieved.
4. The attribute can then be used in a SAML Authorization Assertion inserted into a message fro consumption by a downstream service.

Flow of request:


1. Request with Basic HTTP authentication details I.e Username/Password
2. Gateway Connects to the Oracle database and authenticates and retrieves user attributes based on search query
Oracle Enterprise Gateway (OEG)
Client
4. Gateway receives sporsponse from Web Service and passes it back to client
OEG Gateway
Oracle 10g XE DataBase
3. GatewayValidates attribute and inserts a SAML Assertion into the message and passes it onto web service.
Web Service Interface

This guide applies to OEG software products, from version 5.1 upwards.

In this guide the Oracle Database Server used is Oracle 10g Express Edition Database.

1.2. Oracle 10g Express Edition Database Architecture

Oracle Database 10g Express Edition (Oracle Database XE) is an entry-level, small-footprint database based on the Oracle Database 10g Release 2 code base that's free to develop, deploy, and distribute; fast to download; and simple to administer. Oracle Database XE is a great starter database for:

* Developers working on PHP, Java, .NET, XML, and Open Source applications
* DBAs who need a free, starter database for training and deployment
* Independent Software Vendors (ISVs) and hardware vendors who want a starter database to distribute free of charge
* Educational institutions and students who need a free database for their curriculum

With Oracle Database XE, you can now develop and deploy applications with a powerful, proven, industry-leading infrastructure, and then upgrade when necessary without costly and complex migrations. Read what users say about Oracle Database XE.

Oracle Database XE can be installed on any size host machine with any number of CPUs (one database per machine), but XE will store up to 4GB of user data, use up to 1GB of memory, and use one CPU on the host machine.

Note: Above information is taken from Oracle Website

1.3. Setup Used for this Guide:

- Oracle Enterprise Gateway
- Oracle 10g Express Edition Database

2.Configuring Oracle 10g Express Edition Database

2.1. The Database

For this guide a sample database populated by users with attributes will be created.
NOTE: Please refer to Oracle documentation for details information for installing and running Oracle 10 Express Edition on Windows and Linux platforms which is available from www.oracle.com

To create this database:

1. A database called 'Employees' with a table called 'employee_data' needs to be created.
2. Once the database Employees has been created the table 'employee_data' needs to be created in this database

3. The script to create the employee_data table:

```
CREATE TABLE employee_data
(
emp_id NUMBER(10, 0) NOT NULL,
username varchar(20),
password varchar(20),
title varchar(30),
age NUMBER(3, 0),
salary NUMBER(10, 0),
perks NUMBER(10, 0),
email varchar(60),
CONSTRAINT employee_pk PRIMARY KEY (emp_id)
);

CREATE SEQUENCE employee_data_SEQ
        INCREMENT BY 1
        START WITH 1
        NOMINVALUE
        NOCYCLE
        ORDER;

CREATE OR REPLACE TRIGGER employee_data_tgr
BEFORE INSERT ON employee_data
FOR EACH ROW
WHEN (new.emp_id IS NULL)
```

```
BEGIN
  SELECT employee_data_SEQ.NEXTVAL
  INTO   :new.emp_id
  FROM   dual;
END;
```

4. To create the sample user data within the employee _data table the following script can be used:

```
INSERT INTO employee_data (username, password, title, age, salary, perks,
email)
values ('Hubertf', 'goodNews', 'VP', 27, 120000, 40000,
'hubertf@planetexpress.com');
INSERT INTO employee_data (username, password, title, age, salary, perks,
email) values ('Johnd', 'badNews', 'Senior Programmer', 32, 120000, 25000,
'john_hagan@planetexpress.com');
INSERT INTO employee_data (username, password, title, age, salary, perks,
email) values ('Ganeshj', 'greatNews', 'Senior Programmer', 32, 110000,
20000, 'g_pillai@planetexpress.com');
INSERT INTO employee_data (username, password, title, age, salary, perks,
email) values ('Anamikar', 'seriousNews', 'Web Designer', 27, 90000, 15000,
'ana@planetexpress.com');
```

5. This will add the user Hubertf and 3 more users to the database.
6. The database is now ready for use.

Summary of Database created:

- **Database Name:** Employees
- **Table Name:** employee_data
- **Column Values:** Username, Password, Title, Salary, Perks and Email Address.

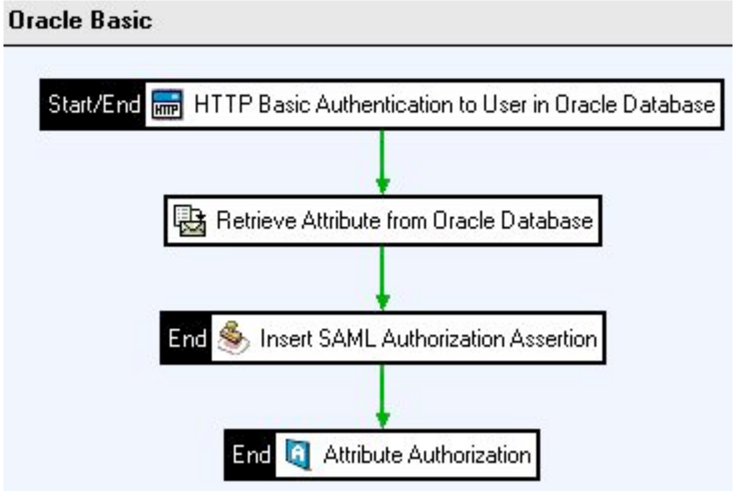
3. Configuring the Gateway

3.1. Create the policy

What the policy will do:

1. Authenticate the user via the Oracle database.
2. Retrieve the Attribute 'title' from the Oracle database.
3. The user will be authorized by verifying that the title of the user is VP using the Attributes filter. If the title is not 'VP' in this case the policy will fail.
4. The attribute will inserted as part of a SAML Authorization Assertion and forwarded to the resource (web service), in this case reflected back to the test client.

The policy will look as follows:


Oracle Basic
Start/End
HTTP Basic Authentication to User in Oracle Database
Retrieve Attribute from Oracle Database
End
Insert SAML Authorization Assertion
End
Attribute Authorization

Creating the policy:

1. Right Click on **Policies** in the tree on the left hand side of Policy Studio
2. Name the Policy "OracleB" for Oracle Basic Policy
3. Click on the Policy and drag a **"HTTP Basic" filter** located in the **"Authentication" group** of the filter palette located on the right hand side of Policy Studio

4. Name of the filter can be left default or changed to any descriptive name.
5. **Realm:** Populated automatically by the value specified in System Settings in the Gateway
6. **Credential Format:** select User Name from the drop down list
7. **Repository Name**: Click on 'Add' to open the Authentication Repository window
8. From the **'Repository Type'** drop down box select **'Database Repositories'.**
9. Name the repository 'OracleEX'.
10. **Database Location:** Click on **'Add'**
11. A window will appear for the configuration of the connection to the database
12. Change the name of the database to 'Employees'
13. **URL:** jdbc:oracle:thin:@ip_of_database_host:1521:xe
14. **Username:** vuser (assuming the admin username for Oracle is vuser)
15. **Password:** vuser (assuming the admin password for Oracle EX is vuser)
16. The rest of the values can be left default with the exception of 'Time between Eviction (ms)' set to 10000
17. Click on **'OK'**
18. Back on the 'Authentication Repository' window click on **'Add/Edit'** next to **'Database Query'**.
19. In the 'Name' field type 'Authentication'
20. The query should look like this:

    SELECT * FROM employee_data WHERE username = ${authentication.subject.id}

    This query will take the user name from the HTTP header and return all the appropriate rows in the database
21. **Statement Type:** Query
22. **Table Structured With:** Attributes as column names
23. Click on **'OK'**
24. In the **'Authentication Repository'** window all options can be left as is under the **'Format password received from client'** section.
25. Under the **'Query result processing'** section it should be configured as follows:

    Password Column: Password

Specify the name of the database table column that contains the user's password. The contents of this column will be compared to the password submitted by the user.

Password Type: Clear

Depending on how the user's password has been stored in the database, select either "Clear Password" or "Digest Password" from the dropdown.

Authorization Attribute Column: username (defines the name of the

By running the **Database Query**, all of the user's attributes are returned. Only the user's username and password are used for the actual authentication event. It is also possible to use one of the other user's attributes for authorization at a later stage in the policy. The additional "authorization attribute" should be either a username or an X.509 distinguished name (DName). The name of the column containing either the username or the DName should be entered here, but only if this value is required for authorization purposes.

Authorization Attribute Format: User Name

The XML Gateway's authorization filters all operate on the basis of a username or DName. In other words, they all evaluate whether a user identified by a username or DName is allowed to access a specific resource. Select the appropriate format from the dropdown depending on what type of user credential is stored in the database table column entered above.

26. Next drag a **'Retrieve from Database' filter** from the **'Attributes' group**

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

28. **User ID:** Change this to: **authentication.subject.id** which now contains the user name as contained in the HTTP authentication header.

    1.

29. **Database Location:** Select the **'Employees Database'** from the list as configured in the 'HTTP Basic' filter.

30. Click on **'OK'**

31. This will then revert back to the main **'Retrieve from Database'** window.

32. Click on **'Add'** next to **'Database Queries'**

33. For this guide we will retrieve the job title of the user, so the name of the query will be **'Get Title for user'**

34. The query itself should look like follows:

    SELECT title from employee_data where username = ${authentication.subject.id}

    The query is looking for the title of the username that has been authenticated

35. **Statement Type:** Query

36. **Table Structured With:** Attributes as column names

37. Click on **'OK'** and then click on **'Finish'**

38. The 'Retrieve from Database' filter should now be configured correctly.

39. The next filter to be configured in the circuit is the **'Attributes' filter** from the **'Authorization' group.**

40. Click on **'Add'** to open the **'Add Attributes'** window.

41. Then click on **'OK'** and the click on **'Finish'**

42. The next filter in the circuit is an **'Insert SAML Authorization Assertion' filter** which can be found in the **'Authorization' group.**

43. **Expire In:** Set to a desired time frame

44. SOAP Actor/Role: select **'Current Actor/Role Only'**

45. Under **'Advanced Options'** select **'Insert SAML Attribute'** statement

46. For 'Resource' enter: ${http.request.uri} (this is the attribute name for the URI on which the HTTP request was received by XML Gateway)

47. For 'Action' any value can be entered. Allow is used for this guide.

48. Click on **'Finish'**

49. The last filter in the circuit is the 'Reflect' filter to reflect the message back to the test client (in this case OEG Service Explorer).

50. Drag the **'Reflect' filter** from the **'Utility' group** in the filter palette.

51. Click on **'OK'**

52. Make sure all filters are connected to one another with a green success path.

53. The policy has now been configured.

3.3. Create a Failure Path for the Policy

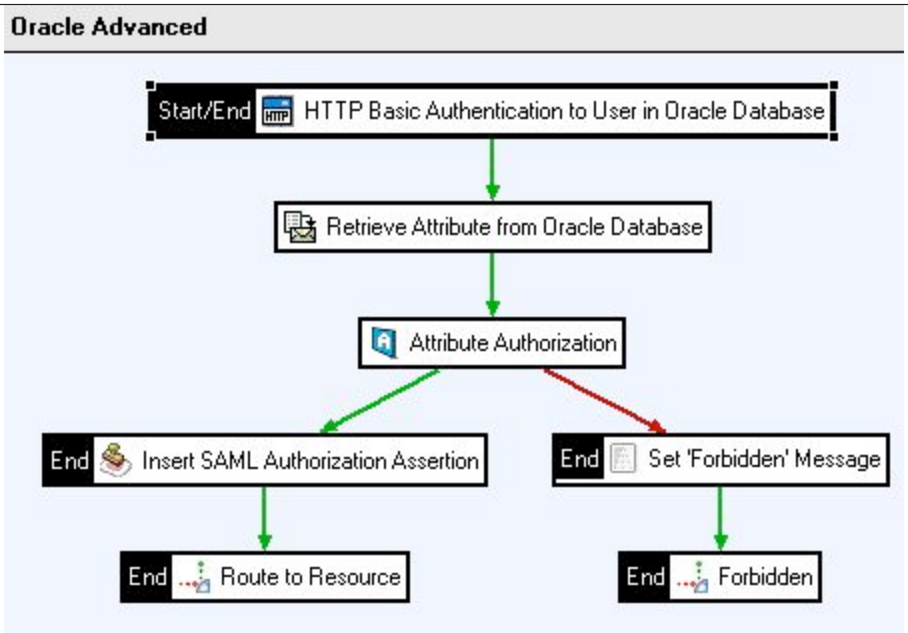
As the current policy will only Authorize users with the ‘title’ of ‘VP’, as failure path will be configured for user who do not have ‘VP’ as a title and are trying to access the web service.

1. Drag a **‘Set Message’ filter** from the **‘Conversion’ group.**
2. For filter name use: Set 'Forbidden' Message
3. Content-Type: text/xml
4. Message Body:

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns:Response xmlns:ns="www.planetexpress">
      <ns:string>Forbidden</ns:string>
    </ns:Response>
  </soap:Body>
</soap:Envelope>
```

5. Click on **‘OK’**
6. The next filter is a **‘Reflect’ filter** from the **‘Utility’ group**.
7. Name the filter ‘Forbidden’
8. Change the **‘HTTP Response code status’** to: **403** (indicating Forbidden resource)
9. Drag a red failure path from the ‘Attribute Authorization filter to the ‘Set Message’ filter. Then connect the ‘Set Message filter to the ‘Reflect’ filter titled ‘Forbidden’.

The policy should now look like this:


Oracle Advanced
Start/End
HTTP Basic Authentication to User in Oracle Database
Retrieve Attribute from Oracle Database
Attribute Authorization
End
Insert SAML Authorization Assertion
End
Set 'Forbidden' Message
End
Route to Resource
End
Forbidden

3.4. Create a New Relative path to point to Policy

- Open **Policy Studio**
- Expand **Processes** and then **Oracle Enterprise Gateway**
- Right Click on **Default Services** and select “Add Relative Path”
- Name the Relative path as follows: /OracleA
- Map the path to the newly created policy titled “Oracle Advanced”
- Click ‘OK’

3.5. Ensure policies are updated on the XML Gateway

- Open the **Policy Studio**
- Click on **Settings**
- Select **Refresh Server** to ensure that the changes made are propagated to the live XML Gateway.

4. Test the Policy with OEG Service Explorer

A test message will now be sent through to test the policy.
The test SOAP message used for this guide is:

```
<soap:Envelope
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns:Response xmlns:ns="www.planetexpress">
      <ns:string>access required</ns:string>
    </ns:Response>
  </soap:Body>
</soap:Envelope>
```

Set up a message in OEG Service Explorer

- Open **OEG Service Explorer**
- Load a message request
- Click on **Settings** just above the **Send Request** button
- Then Click on **Connection Settings**
- Make sure that the URL is set correctly. In this case it will be http://ip_of_gateway:8080/OracleB for the basic and also http://localhost:8080/OracleA for the advanced policy
- Click on **OK**
- Click on the **HTTP Authentication** tab
- Select **HTTP Basic**
- Enter the **Username** and **Password** of the user that will be Authenticated via LDAP
- User Credentials:
  User: hubertf
  Password: goodNews
- Click on **Finish**
- The message would now have been sent through

5. Conclusion

This document is a simplistic demonstration on how to setup the Oracle Enterprise Gateway to authenticate, retrieve and use attributes located in an Oracle 10g Express Edition database.
This configuration can be part of a larger policy, including features such as XML threat detection and conditional routing, features which are out of scope in this document but are covered in other documents which can be obtained from the Oracle at http://www.oracle.com.

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

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

SOFTWARE. HARDWARE. COMPLETE.