Oracle Identity Analytics System Integrator's Guide

11*g* Release 1

Part No: E23129–01
April 2011

110510@25097

Contents

Preface

About This Guide

This guide describes how to integrate Oracle® Identity Analytics 11gR1 software with other applications in a heterogeneous IT environment. Included in this guide is information about how to integrate with Oracle Identity Manager, which is Oracle's resource provisioning solution.

Who Should Read This Guide

The *Oracle Identity Analytics 11gR1 System Integrator's Guide* is written for deployment engineers and service providers who are responsible for integrating Oracle Identity Analytics with other IT systems.

- System administrators and service providers who need information about how to monitor and administer the Oracle Identity Analytics software at a systems level should see the *Oracle Identity Analytics 11gR1 System Administrator's Guide*.
- Compliance officers and IT specialists who need to configure and maintain role management and compliance functionality should see the *Oracle Identity Analytics 11gR1 Business Administrator's Guide*.
- Business managers and other users in a supervisory role who need information about how to use the Oracle Identity Analytics 11gR1 software to grant employees and partners access to applications, check for access violations, and so on should see the *Oracle Identity Analytics 11gR1 User's Guide*.

CHAPTER 1

Integrating With Oracle Identity Manager, Preferred Method

Introduction

This section describes how to configure Oracle Identity Analytics (OIA) and Oracle Identity Manager (OIM) so that the two products can be used together. This newer, preferred integration method uses database imports for users and accounts, which allows for incremental imports from Oracle Identity Manager. To use this integration method you must have at least Oracle Identity Manager version 11gR1 BP3 or version 9.1.0.2 BP14a, and at least Oracle Identity Analytics 11gR1 BP3. To integrate older versions of Oracle Identity Manager and Oracle Identity Analytics, see the Chapter 2, “Integrating With Oracle Identity Manager, Deprecated Method,” documentation in the next chapter.

Overview

Oracle Identity Analytics software and Oracle Identity Manager (OIM) software work together seamlessly when integrated using the Thor-API connection mechanism. When integrated, Oracle Identity Manager serves as the automated provisioning and identity synchronization solution, while Oracle Identity Analytics defines the Role-based Access Control (RBAC) framework, the attestation process, and the approach to Segregation of Duties (SoD) policy enforcement. Rather than assigning individual access entitlements, the RBAC framework allows organizations to assign and unassign roles as a means of controlling user access on various applications.

In a fully-integrated scenario, provisioning and role management works in the following manner:

- OIM is the authoritative source for users, accounts, and entitlements. Any update made to the users or their corresponding accounts is done in OIM.

- Oracle Identity Analytics is the authoritative source for role management and role membership. Oracle Identity Analytics is also the authoritative source for policy entitlement definitions. (*Roles* in Oracle Identity Analytics correspond to *roles* in OIM 11.*x*, and *groups* in OIM 9.*x*. Further, *policies* in Oracle Identity Analytics correspond to *access policies* in OIM.)
- All roles are defined and created in Oracle Identity Analytics. All entitlements for policies and role-to-user relationships are managed from Oracle Identity Analytics.
- Role, Policy, and Role-Membership updates should no longer be made in Oracle Identity Manager.

Understanding Terminology in Oracle Identity Analytics and Oracle Identity Manager

The following table maps Oracle Identity Analytics terminology to Oracle Identity Manager terminology.

Oracle Identity Analytics Terminology	Oracle Identity Manager Terminology
Resource Type	Resource Object
Resource Type Attributes (NameSpace Attributes)	Provisioning Attributes and Entitlements
Resource	IT Resource
Global Users	Users or Xellerate End Users
Roles	Groups (9.x) / Roles (11g)
Policies	Access Policies

To Configure Oracle Identity Analytics and Oracle Identity Manager to Work Together (Preferred Integration Method)

Before You Begin -

- **At least Oracle Identity Manager version 11gR1 BP3 or version 9.1.0.2 BP14a is required.**
- **At least Oracle Identity Analytics 11gR1 BP3 is required.**
- Both Oracle Identity Manager and Oracle Identity Analytics should be installed on servers running the same version of the application server software, as well as the same version of the Java® Virtual Machine (JVM).

1. Copy the required Oracle Identity Manager API JAR files to Oracle Identity Analytics.

See "Step 1: Copy the Required Files From the OIM Server" on page 11

2. In Oracle Identity Analytics, edit the required and optional configuration files.

   See "Step 2: Edit the Oracle Identity Analytics Configuration Files" on page 12
3. In Oracle Identity Manager, log on to the Design Console and edit the required forms.

   See "Step 3: Modify the Oracle Identity Manager Forms Using the Form Designer" on page 13
4. In Oracle Identity Manager, configure the data collection scheduler.

   See "Step 4: Configure the Oracle Identity Manager Data Collection Scheduler" on page 15
5. In Oracle Identity Analytics, create a connection to Oracle Identity Manager. Establish a connection by entering authentication details.

   See "Step 5: Configure Oracle Identity Analytics to Connect to Oracle Identity Manager" on page 16
6. In Oracle Identity Analytics, import data from Oracle Identity Manager.

   See "Step 6: Import the Oracle Identity Manager (OIM) Data Into Oracle Identity Analytics (OIA)" on page 17
7. To send real time changes from Oracle Identity Analytics to Oracle Identity Manager, change the Oracle Identity Analytics configuration files related to workflows.

   See "Step 7: Configure the Oracle Identity Analytics (OIA) Workflows to Export Data to Oracle Identity Manager (OIM)" on page 23
8. In Oracle Identity Manager, review automatic role assignment and role management.

   See "Step 8: Review Oracle Identity Manager Automatic Role Assignment and Role Management Settings" on page 24

Step 1: Copy the Required Files From the OIM Server

- Copy the following Oracle Identity Manager Java API JAR files located in the <*OIMDesignConsole*> `/lib` folder to the Oracle Identity Analytics `$RBACX_HOME/WEB-INF/lib` folder:
  - `xlAPI.jar`
  - `xlCache.jar`
  - `xlDataObjectBeans.jar`
  - `xlDataObjects.jar`
  - `xlScheduler.jar`
  - `xlUtils.xls`
  - `xlVO.jar`
- Copy the following JAR files located in the <*IDM-HOME*> `/server/lib` folder to the Oracle Identity Analytics `$RBACX_HOME/WEB-INF/lib` folder:
  - `xlCrypto.jar`

  - `wlXLSecurityProviders.jar`
  - `xlAuthentication.jar`
  - `xlLogger.jar`
- Copy the `conf` folder from <*OIMDesignConsole*>`/conf` to the Oracle Identity Analytics `$RBACX_HOME/WEB-INF/lib` folder.
- If using at least Oracle Identity Manager 11gR1 BP3, also copy the following OIM files :
  - `oimclient.jar`

    Use the version located in the <*OIMDesignConsole*>`/lib` folder. (**Important** — Do not use a copy of this JAR file located in any other directory.)
  - `iam-platform-utils.jar`

    This file is located in the <*OIMDesignConsole*>`/lib` folder.
- If deploying to a JBoss application server, copy `jbossall-client.jar`
- If deploying to a WebLogic application server, *and if Oracle Identity Analytics and Oracle Identity Manager are on different WebLogic domains*, copy <*OIMDesignConsole*>`/ext/wlfullclient.jar`

  Note - If `wlfullclient.jar` is not present in Oracle Identity Manager, follow these steps to generate it:

  1. Type `cd` <*WLS-HOME*>`/server/lib` , where <*WLS-HOME*> is the base WebLogic installation directory
  2. Type `java -jar wljarbuilder.jar`
  3. Copy the `wlfullclient.jar` file to the `$RBACX_HOME/WEB-INF/lib` folder

Step 2: Edit the Oracle Identity Analytics Configuration Files

1. Enable Oracle Identity Manager as a supported provisioning server by editing `iam-context.xml` in the `RBACX_Home/WEB-INF` folder as follows:

   a. Uncomment the following lines at the start of `iam-context.xml`:

```
<import resource="oim-commons-context.xml"/>
<import resource="oim-11g-context.xml"/>  <!-- This also works with at least Oracle  /
Identity Manager 9.1.0.2 BP14a-->
```

   b. Enable the following:

```
<entry key="oracle">
  <ref bean="oimSolution"/>
</entry>
```

2. (Optional) To map Oracle Identity Manager extended attributes to Oracle Identity Analytics custom properties, add the following mappings to `oim-commons-context.xml` as appropriate:

- For Users, complete the mapping by updating the `value` attribute with the Oracle Identity Manager extended attribute name, as follows:

  ```
  <util:map id="iamUserToUserCustomProperties">
          <!--entry key="customProperty1" value="usr_udf_cust1"/>
          <entry key="customProperty2" value="usr_udf_cust2"/>
          <entry key="customProperty19" value="usr_udf_cust19"/-->
  </util:map>
  ```

- For Roles, complete the mapping by updating the `value` attribute with the Oracle Identity Manager extended attribute name, as follows:

  ```
  <util:map id="iamRoleCustProperties">
          <!--entry key="customProperty1" value="Groups.Group Name"/-->
  </util:map>
  ```

3. (Optional) If enabling closed-loop remediation, edit `oim-11g-context.xml` and add the appropriate mappings as follows:

   ```
   <property name="accountIdentifierMap">
               <map>
                   <entry key="AD User" value="UD_ADUSER_UID"/>
               </map>
   </property>
   ```

4. Edit `$RBACX_HOME/conf/oimjdbc.properties`. This should contain the Oracle Identity Manager database information.

   a. Run the `encryptPassword` tool in the `samples` folder to encrypt the database password located in the `oimjdbc.properties` file.

   b. Open the `oim-11g-context.xml` file for editing and search for the word *password*.

   c. Update the XML so that the tags look like the following sample:

   ```
       <prop key="user">${oim.jdbc.username}</prop>
       <!--prop key="password">${oim.jdbc.password}</prop-->
       <prop key="password">${oim.jdbc.password.encrypted}</prop>
       <prop key="SetBigStringTryClob">true</prop>
     </props>
   ```

   d. Save your changes.

Step 3: Modify the Oracle Identity Manager Forms Using the Form Designer

In this step you will open Form Designer and, for each OIM resource, add the three properties that OIA needs to exchange data with OIM.

1. Log in to the Oracle Identity Manager Design Console.
2. Open the Form Designer.
3. For each Resource, the following properties need to be added to some identified feed for accounts, policies, and entitlements imports:
   - **AccountName** - Identifies the unique account in the target system

- **ITResource** - Identifies the unique IT Resource field for the target system
- **Entitlement** - Identifies the account attribute designated for privileges
- **OIAParentAttribute** - Add this property only if you have installed at least **OIM 9.1.0.2 BP14a**. This property identifies the parent or mandatory entitlement attributes.

  Complete this step as follows:

  a. Locate the Process Form for the given resource.

     Note - The `AccountName` and `ITResource` properties are on the parent form, and the `Entitlement` and `OIAParentAttribute` properties are on the child form.

  b. Open the child Process Form and create a new version.

  c. Click the **Properties** tab.

  d. Locate *ONLY ONE* entitlement field per form, click **Add Property**, and add the `Entitlement = true` property setting.

     If there are multiple Entitlement child forms, add one `Entitlement = true` property setting per Entitlement form.

  e. If you have installed at least **OIM 9.1.0.2 BP14a** (but not including OIM 11gR1 and higher), do the following: Locate *ONLY ONE* entitlement field per form, click **Add Property**, and add the `OIAParentAttribute = true` property setting.

     If there are multiple Entitlement child forms, add one `OIAParentAttribute = true` property setting per Entitlement form.

  f. Save the child form and make it active.

  g. Locate the parent process form and create a new version.

  h. Click the **Properties** tab.

  i. Locate the field that uniquely identifies the account in the target system, click **Add Property**, and add the `AccountName = true` property setting.

     See the following screen capture for an example.

  j. Locate the ITResource field for the target system, click **Add Property**, and add the `ITResource = true` property setting.

  k. Save the parent form and make it active.

4. Repeat for each Resource.
5. Restart the Oracle Identity Analytics server.

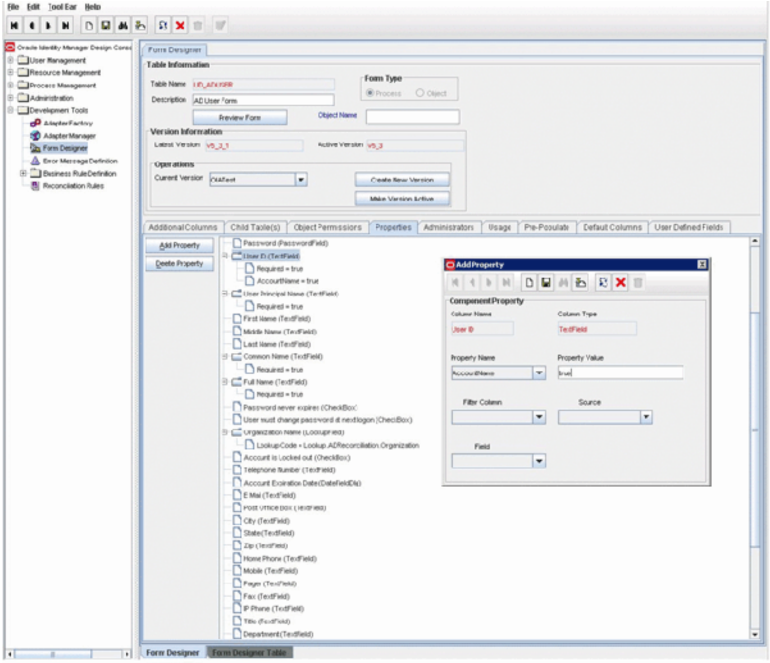


Table Name: UD_ADUSER
Description: AD User Form
Latest Version: v5_3_1
Active Version: v5_3
Current Version: OIATest

Column Name: User ID
Column Type: TextField
Property Name: AccountName
Property Value: true

Step 4: Configure the Oracle Identity Manager Data Collection Scheduler

Use the following steps to register the Oracle Identity Manager scheduled task that is required to support the OIA-OIM integration.

Before You Begin - Verify that the OIM installation/upgrade script created the *DataCollection Schedule Job* in OIM and that the job is enabled but not scheduled for execution. Your integration will not work without this important job.

Follow these steps to register the task with OIM:

1. Export the `task.xml` file from the MDS.

   The MDS path for `task.xml` is `/db/task.xml`.
2. Open the `task.xml` file for editing.
3. Add the following scheduled task to the `task.xml` file and save the file.

```
<task>
    <name>DataCollection Schedule Task</name>
    <class>com.thortech.xl.schedule.tasks.DataCollectionTask</class>
    <description>DataCollection Schedule Tasks</description>
    <retry>5</retry>
</task>
```

4. Reimport `task.xml` into the MDS so that the scheduled task is available for creating the data collection scheduled job.
5. Enable the DataCollection Schedule Task if you are using Oracle Identity Manager 9.1.0.2.

   If you are using at least Oracle Identity Manager 11gR1, the DataCollection Schedule Task is already enabled so you should skip this step.

Step 5: Configure Oracle Identity Analytics to Connect to Oracle Identity Manager

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Provisioning Servers.
4. Click New Provisioning Server Connection.

   The New Provisioning Server Connection wizard asks you to choose the type of provisioning server connection that you want to create.
5. From the Type of Provisioning Server Connection drop-down menu, select Oracle and click Next.
6. Complete the form:
   - **Server Name** - Type the Oracle Identity Manager server name.
   - **Xellerate Home** - Type the path to the `xellerate` folder in OIM. (Example: `C:oraclexellerate`

     If Oracle Identity Manager is on a separate machine, create a local `xellerate` folder and copy the `config` folder from `<OIMDesignConsole>` in the `xellerate` folder.
   - **Login Config** - Type the path to the authentication configuration (`auth.config`) file. (Example: `C:oraclexellerateconfigauthwl.conf`)
   - **Provider URL** - Type the provider URL. The format for this field is as follows:
     - **WebLogic** -

       `t3://host:7001`
     - **JBoss** -

       `jnp://host:1099` (The default port number in a clustered environment is 1100.)
     - **WebSphere** -

       `corbaloc:iiop:host:2809`
   - **Initial Context Factory** - Enter the name of the environment property for specifying the initial context factory. The default values are as follows:
     - **WebLogic** -

       `weblogic.jndi.WLInitialContextFactory`

    - **JBoss** -

      `org.jnp.interfaces.NamingContextFactory`

    - **WebSphere** -

      `com.ibm.websphere.naming.WsnInitialContextFactory`

  - **User Name** - Enter the OIM user name. (example: `xelsysadm`) The specified OIM user needs to have system administrator priviliges.
  - **Password** - Enter the OIM password.

7. Click Save.

Step 6: Import the Oracle Identity Manager (OIM) Data Into Oracle Identity Analytics (OIA)

Complete this step if you have data in Oracle Identity Manager that you want to use to populate the Oracle Identity Analytics Identity Warehouse. Importing data about Users, Resources, Entitlements, and so on, eliminates the need to manually create this information in Oracle Identity Analytics.

Note - Importing data from Oracle Identity Manager into Oracle Identity Analytics using this procedure should be a one-time event that takes place when first configuring the systems.

Schedule or run the import jobs in the following order:

1. Import **Resource Metadata**. See *To Import Resource Metadata* for details.
2. Import **Resources**. See “To Import Resources” on page 18 for details.
3. Import the **Glossary** Data. See “To Import Glossary Data” on page 19 for details.
4. Import **Entitlements, Users, and Accounts**. See “To Import Entitlements, Users, and Accounts” on page 19 for details.
5. Import **Policies**. See “To Import Policies” on page 20 for details.
6. Import **Roles**. See “To Import Roles” on page 21 for details.

To Import Resource Metadata

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Resource Metadata.

   The next page will prompt you to choose the resource from the list of available resources for which metadata on attributes needs to be imported.
5. Select the specific resource type.

6. Under Data Selection Source, select the appropriate Connection Name and click Next.
7. Complete the form by entering the Name and Description of the Job.
8. Choose one of the following:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
9. Click Finish to generate the Import Job.

   The import resource metadata job runs on the scheduled date and time.
10. Set (or validate) the parent attribute for each attribute category by following the steps in the "To Validate That the Parent Attribute for Each Attribute Category is Set" on page 21 topic.

    If you are running OIA BP3 and OIM 9.1.0.2 BP14a, *or* if you are running OIA BP4 and at least OIM 11gR1 BP3, you need to complete this step to set the parent attribute for each attribute category.

    Otherwise, complete this step as a validation step to verify that the parent attribute for each attribute category has been set appropriately.
11. Verify that the resource metadata was properly imported into Oracle Identity Analytics either by accessing the Oracle Identity Analytics Resources Types tab (choose Configuration > Resources Types), or by following the steps in the "To Verify That Each Import Job Completed Successfully" on page 21 topic.

To Import Resources

Note - An *ITResource* in OIM corresponds to a Resource in Oracle Identity Analytics.

1. If necessary, log in to Oracle Identity Analytics, choose Administration > Configuration, and click Import/Export.
2. To start the import resources job, choose Schedule Job > Import > Import Resources.
3. Under Data Selection Source, select the appropriate Connection Name and click Next.
4. Complete the form by typing a name and description for the job.
5. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
6. Click Finish to generate the import job.

   The import resources job runs on the scheduled date and time.
7. Verify that the resources are imported into Oracle Identity Analytics from Identity Manager either by accessing the Oracle Identity Analytics Resources tab (choose Identity Warehouse > Resources), or by following the steps in the "To Verify That Each Import Job Completed Successfully" on page 21 topic.

To Import Glossary Data

1. If necessary, log in to Oracle Identity Analytics, choose Administration > Configuration, and click Import/Export.
2. To start the import glossary job, choose Schedule Job > Import > Import Glossary.
3. Under Data Selection Source, select the appropriate Connection Name and click Next.
4. Complete the form by typing a name and description for the job.
5. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
6. Click Finish to generate the import job.

   The import glossary job runs on the scheduled date and time.
7. Verify that the glossary data imported into Oracle Identity Analytics from Identity Manager either by following the steps in the "To Verify That Each Import Job Completed Successfully" on page 21 topic.

To Import Entitlements, Users, and Accounts

1. If necessary, log in to Oracle Identity Analytics, choose Administration > Configuration, and click Import/Export.
2. To start a new import job, choose Schedule Job > Import > Import Entitlements, Users, and Accounts.
3. Under Data Selection Source, select the appropriate Connection Name and click Next.
4. Choose one of the following:
   - **Load all resources defined in the system at the time the job is run** - Choose this option to import data from all resources.
   - **Load only those resources selected in the table** - Choose this option to import data only from select resources. If you choose this option, select one or more resources in the table.
5. Complete the form as follows:
   a. Type a name and description for the job.

   b. In the **Data to Load** section, select the **Entitlements** option if, in addition to accounts and users, you also want to import the users' entitlements data. Otherwise, clear the **Entitlements** option box and only the accounts and users data will be imported.

   c. In the **Import Type** section, choose one of the following:
      - **Complete** - All entities found on the OIM server will be imported.

- **Incremental** - All OIM entities updated since the last successful import will be imported.

d. Choose one of the following:

- To run the job immediately, select the Run the Job Now option.
- To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.

6. Click Finish to generate the import job.

   The import job runs on the scheduled date and time.

7. Verify that the entitlements, users, and accounts are imported into Oracle Identity Analytics from Identity Manager either by accessing the Users View in Oracle Identity Analytics (choose Identity Warehouse > User), or by following the steps in the "To Verify That Each Import Job Completed Successfully" on page 21 topic.

To Import Policies

Note - In OIA, a policy represents a specific privilege on a specific resource, whereas in OIM, a single access policy can represent multiple resources. Consequently, when importing an OIM policy that represents multiple resource types, OIA will create multiple policy instances (one policy instance per resource) and save the policy with the resource name appended to the policy name. Going forward, Oracle recommends that you not assign more than one resource to a policy in OIM.

1. If necessary, log in to Oracle Identity Analytics, choose Administration > Configuration, and click Import/Export.
2. To start the import policies job, choose Schedule Job > Import > Import Policies.
3. Under Data Selection Source, select the appropriate Connection Name and click Next.
4. Complete the form by typing a name and description for the job.
5. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
6. Click Finish to generate the import job.

   The import policies job runs on the scheduled date and time.

7. Verify that the policies are imported into Oracle Identity Analytics from Identity Manager either by accessing the Oracle Identity Analytics Policies tab (choose Identity Warehouse > Policies), or by following the steps in the "To Verify That Each Import Job Completed Successfully" on page 21 topic.

To Import Roles

Note - Groups defined in OIM are imported as Roles within Oracle Identity Analytics. In addition, the OIM Group-to-Access-Policy relationship is imported as a Roles-Policy relationship in Oracle Identity Analytics. For the import to work, you should have already successfully completed a Policy import.

In addition, the OIM Group-User relationship is imported and recreated as a Role-User relationship in Oracle Identity Analytics. To establish the Role-User relationship, verify that you have already imported Users.

1. If necessary, log in to Oracle Identity Analytics, choose Administration > Configuration, and click Import/Export.
2. To start the import roles job, choose Schedule Job > Import > Import Roles.
3. Under Data Selection Source, select the appropriate Connection Name and click Next.
4. Complete the form by typing a name and description for the job.
5. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
6. Click Finish to generate the import job.

   The import resources job runs on the scheduled date and time.
7. Verify that the roles are imported into Oracle Identity Analytics from Identity Manager either by accessing the Oracle Identity Analytics Roles tab (choose Identity Warehouse > Resources), or by following the steps in the "To Verify That Each Import Job Completed Successfully" on page 21 topic.

To Verify That Each Import Job Completed Successfully

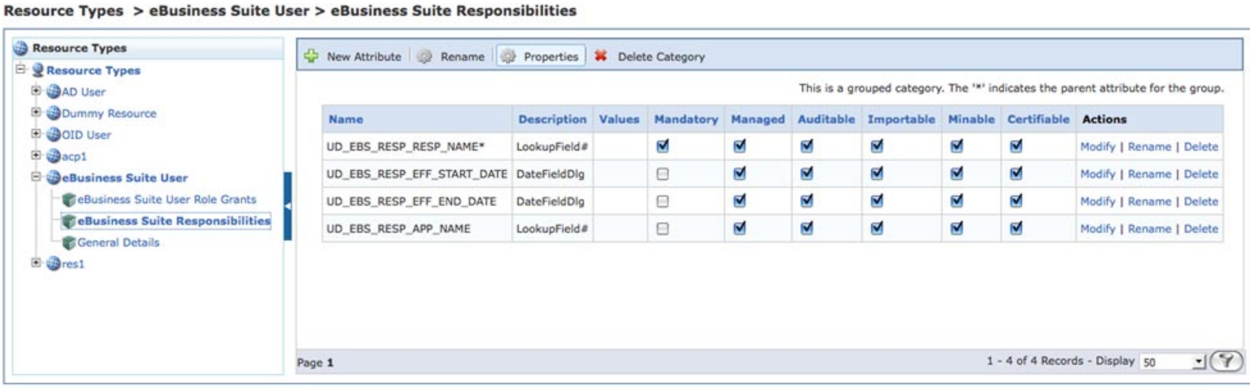
1. Log in to Oracle Identity Analytics.
2. Choose Administration > Auditing & Events.
3. Click Import/Export Logs.
4. In the table, find the entries for your import jobs.
5. Click the entry in the **Description** column to view the Import Log Details page.
6. Verify that the number or Oracle Identity Manager export records (Number of Output Records) and the number of Oracle Identity Analytics import records (Number of Input Records) are the same.

To Validate That the Parent Attribute for Each Attribute Category is Set

After Importing Resource Metadata, complete this step as a validation step to verify that the parent attribute for each attribute category has been set appropriately.

Note - This procedure is required if you are running OIA BP3 and OIM 9.1.0.2 BP14a, *or* if you are running OIA BP4 and at least OIM 11gR1 BP3. Follow these steps to manually assign the parent attribute for each attribute category.

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Resource Type.
4. Click the + for each namespace to see the attribute categories for the selected Resource Type.
5. Click an attribute category. Attribute categories correspond to the child forms in OIM.



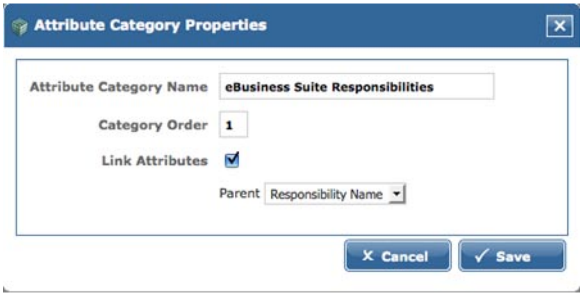
This is a grouped category. The '*' indicates the parent attribute for the group.

Name	Description	Values	Mandatory	Managed	Auditable	Importable	Minable	Certifiable	Actions
UD_EBS_RESP_RESP_NAME*	LookupField#		☑	☑	☑	☑	☑	☑	Modify | Rename | Delete
UD_EBS_RESP_EFF_START_DATE	DateFieldDlg		□	☑	☑	☑	☑	☑	Modify | Rename | Delete
UD_EBS_RESP_EFF_END_DATE	DateFieldDlg		□	☑	☑	☑	☑	☑	Modify | Rename | Delete
UD_EBS_RESP_APP_NAME	LookupField#		□	☑	☑	☑	☑	☑	Modify | Rename | Delete

6. Click **Properties** in the menu.

   The Attribute Category Properties dialog box opens.

7. Do the following:
   - Verify that the **Link Attributes** option is selected and that the **Parent** list is set to the field that was marked as the `OIAParentAttribute` in OIM.
   - If the correct field is not selected, choose the correct parent attribute from the **Parent** list and click **Save**.



Attribute Category Name: eBusiness Suite Responsibilities

Category Order: 1

Link Attributes: ☑

Parent: Responsibility Name

Step 7: Configure the Oracle Identity Analytics (OIA) Workflows to Export Data to Oracle Identity Manager (OIM)

This section describes how to configure workflows to export data in near real-time from Oracle Identity Analytics (OIA) to Oracle Identity Manager (OIM). As noted earlier, all roles are defined and created in Oracle Identity Analytics. Hence, Oracle Identity Analytics is the authoritative source for role management, role membership, and policy entitlement definitions.

For information about closed loop compliance, see the "Oracle Identity Analytics Web Services" on page 59 section.

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Workflows.

   A list of workflows displays.
4. The following three configuration files need to be modified:
   - Role-creation
   - Role-modification
   - Role-membership

     Modify each configuration file as follows:
5. Click the workflow name.
   a. In the **Steps** table, scroll down and click the **Finish** step.

      The Edit Workflow Step page opens.

   b. Click **Add Pre-Functions**

      The Pre-Functions pop-up opens.

   c. In the pop-up, select "Export IAM Role Function."

   d. Choose the Oracle Identity Manager connection name that you created previously.

e. Click Save.

Repeat these steps until the Role-creation, Role-modification, and Role-membership workflows have been modified.

Step 8: Review Oracle Identity Manager Automatic Role Assignment and Role Management Settings

When integrating with Oracle Identity Analytics, Oracle recommends that you no longer use OIM Automatic Role Assignment and Role Management.

To Migrate From the Deprecated OIM-OIA Integration to the Preferred OIM-OIA Integration

If you have an older integration, the following steps must be performed before using the Oracle Identity Analytics 11gR1 BP3 release. Otherwise, your data will be corrupted and you will end up with many unusable objects in the system.

Before You Begin - Synchronize your Oracle Identity Manager data with your Oracle Identity Analytics data. This step is important!

1. In Oracle Identity Analytics, rename the namespace names from the "Resource Type" names in OIM to the "Resource Object" names in OIM.

   Note - You will need to perform the following steps for each OIA namespace that is synchronized with the OIM namespace.

   a. Log in to Oracle Identity Analytics.

   b. Choose Administration > Configuration.

   c. Click Resource Types.

   d. Select the namespace in the tree on the left side of the page, then click **Rename**.

   e. Type the new value in the pop-up.

   Refer to the `iam-context.xml` file in your OLDER installation, and go to the section with the namespaceMap:

```
<property name = "namespaceMap">
    <map>
        <entry key = "AD Server">
            <value>AD User</value>
        </entry>
    </map>
</property>
```

Previously, the namespace name in Oracle Identity Analytics used to be `AD Server`, which corresponds to the `key` value. For the new integration to work, the namespace name in OIA should be `AD User`, which is present in the `value` element.

f. Repeat these steps to manually replace the key with the OIA value for each namespace specified in the older `iam-context.xml` file.

2. Import your Oracle Identity Manager data into Oracle Identity Analytics.

   This step is required because some minor changes need to be imported into OIA.

   Going forward, the way data is represented (accounts and policies, especially) can be updated and maintained.

Understanding Closed Loop Compliance

With the integration of Oracle Identity Analytics and Oracle Identity Manager, it is possible to directly revoke roles and entitlements from Oracle Identity Analytics if the results of the certification process require it. This integration eliminates the need for manual de-provisioning of access for managed resources. In addition, roles and entitlements can still be manually revoked by leveraging the information stored in the remediation configuration module. This takes into account non-managed applications.

If certification remediation is enabled, changes are propagated to Oracle Identity Manager either when the certification is complete, or when the certification end-date is reached (depending on configuration). OIM revokes or re-provisions target system accounts based on the revocations and certifications that occurred during the certification process.

To Configure Resources in Oracle Identity Analytics for Remediation

Every resource type in Oracle Identity Analytics can be separately configured for automatic or manual remediation.

1. Log in to Oracle Identity Analytics.
2. Choose Identity Warehouse> Resources.
3. Click the resource for which remediation action needs to be configured, and go to the Remediation tab.
4. Select the Select Provisioning Mode check box.
5. Choose the mode of provisioning desired for the particular resource.
   - **Auto** - Automatically send role/entitlement updates linked with this resource to Oracle Identity Manager.

     Select the appropriate connection name of the provisioning server and save the changes.

- **Manual** - Use the manual steps for revocation of roles and entitlements using a text editor.

  List the steps to be followed for non-managed system remediation and save the changes.

To Configure Certifications in Oracle Identity Analytics for Remediation

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Identity Certification.
4. Expand the Revoke and Remediation section, and, under the Remediation section, choose one of the following options:
   - **Display Remediation Instructions** - Select to display instructions about how to perform manual remediation of nonmanaged resources.
   - **Perform Closed Loop Remediation on** - Select to specify that the remediation be completed by either the Certification End Date or the Certification Completion Date.

Scheduling Incremental Updates of Users, Accounts, and Entitlements

The OIM-OIA Preferred Integration Method allows for incremental imports of users, accounts, and entitlements from Oracle Identity Manager. Scheduled imports of entitlements, users, and accounts are initially configured as part of the OIM-OIA configuration process. (See "To Import Entitlements, Users, and Accounts" on page 19 in the Chapter 1, "Integrating With Oracle Identity Manager, Preferred Method," section for more information.) Use the steps in this section to schedule additonal imports, or to change an existing scheduled import.

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Entitlements, Users, and Accounts.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Choose one of the following:
   - **Load all resources defined in the system at the time the job is run** - Choose this option to import data from all resources.

- **Load only those resources selected in the table** - Choose this option to import data only from select resources. If you choose this option, select one or more resources in the table.

7. Complete the form as follows:

   a. Type a name and description for the job.

   b. In the **Data to Load** section, select the **Entitlements** option if, in addition to accounts and users, you also want to import the users' entitlements data. Otherwise, clear the **Entitlements** option box and only the accounts and users data will be imported.

   c. In the **Import Type** section, choose one of the following:

      - **Complete** - All entities found on the OIM server will be imported.
      - **Incremental** - All OIM entities updated since the last successful import will be imported.

   d. Choose one of the following:

      - To run the job immediately, select the Run the Job Now option.
      - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.

        Note - For help understanding cron expressions, see *Oracle Identity Analytics Scheduling* in the *Oracle Identity Analytics System Integrator's Guide*.

8. Click Finish to generate the import job.

   The import job runs on the scheduled date and time.

9. Verify that the entitlements, users, and accounts are imported into Oracle Identity Analytics from Identity Manager either by accessing the Users View in Oracle Identity Analytics (choose Identity Warehouse > User), or by following the steps in the "To Verify That Each Import Job Completed Successfully" on page 21 topic.

Troubleshooting

Problem: When OIA tries to connect to OIM, the following error is returned:

```
Illegal Argument Exception thrown ( No Configuration was registered that can
handle the configuration named "xellerate" )
```

Solution: Manually set the security property `auth.login.conf` through `JAVA OPTIONS` before starting the application server.

```
JAVA_OPTIONS="-Djava.security.auth.login.config= /..path../config/authwl.conf
```

Problem:

When starting OIA, the following error is returned:

```
Caused By: java.lang.LinkageError: loader constraint violation: loader (instance
of weblogic/utils/classloaders/ChangeAwareClassLoader) previously initiated
loading for a different type with name "javax/xml/namespace/QName"
```

Solution:

If Oracle Identity Analytics and Oracle Identity Manager are deployed to the same WebLogic domain, remove the `wlfullclient.jar` file from the Oracle Identity Analytics `$RBACX_HOME/WEB-INF/lib` folder. This file is only required if Oracle Identity Analytics and Oracle Identity Manager are on different WebLogic domains. The `wlfullclient.jar` file allows client applications, such as Oracle Identity Analytics, to communicate with the WebLogic Server over the T3 protocol.

Problem:

The following exception is received during integrated operations:

```
java.lang.NoClassDefFoundError:oracle/iam/platform/OIMClient at
Thor.API.tcUtilityFactory.<init>(tcUtilityFactory.java:154) at
com.vaau.rbacx.iam.oracle.OIMIAMSolution.getUtilityFactory(OIMIAMSolution.java:2595)
at com.vaau.rbacx.iam.oracle.OIMIAMSolution.readUsers(OIMIAMSolution.java)
```

Solution:

Copy the following 11g Oracle Identity Manager Java API JAR file to the
`$OIA-HOME/WEB-INF/lib`

folder in Oracle Identity Analytics:

<OIMDesignConsole> `/lib/oimclient.jar`

Problem:

The following error is received during integrated operations:

```
Caused by: java.lang.NoClassDefFoundError: com/thortech/util/logging/Logger at
Thor.API.tcUtilityFactory.<clinit>(tcUtilityFactory.java:80) at
com.vaau.rbacx.iam.oracle.OIMIAMSolution.getUtilityFactory(OIMIAMSolution.java:2595)
at com.vaau.rbacx.iam.oracle.OIMIAMSolution.readUsers(OIMIAMSolution.java:770)
at
com.vaau.rbacx.iam.service.impl.RbacxIAMServiceImpl.importUsers(RbacxIAMServiceImpl.java:1
```

Solution:

Copy the OIM 11g logger JAR file `xlLogger10g.jar` to `$OIA-HOME/WEB-INF/lib`

CHAPTER 2

Integrating With Oracle Identity Manager, Deprecated Method

Introduction

This chapter describes the original approach to configuring Oracle Identity Analytics and Oracle Identity Manager so that the two products can be used together. This older integration method does not support incremental user and account imports. To use this integration method you must have at least Oracle Identity Manager version 9.1.0.2 BP5, and Oracle Identity Analytics 11gR1. A newer, preferred integration method is available that does support incremental user and account imports. For details, see the Chapter 1, "Integrating With Oracle Identity Manager, Preferred Method," documentation in the previous chapter.

Overview

Oracle Identity Analytics software and Oracle Identity Manager (OIM) software work together seamlessly when integrated using the Thor-API connection mechanism. When integrated, Oracle Identity Manager serves as the automated provisioning and identity synchronization solution, while Oracle Identity Analytics defines the Role-based Access Control (RBAC) framework, the attestation process, and the approach to Segregation of Duties (SoD) policy enforcement. Rather than assigning individual access entitlements, the RBAC framework allows organizations to assign and unassign roles as a means of controlling user access on various applications.

In a fully-integrated scenario, provisioning and role management works in the following manner:

- OIM is the authoritative source for users, accounts, and entitlements. Any update made to the users or their corresponding accounts is done in OIM.
- Oracle Identity Analytics is the authoritative source for role management and role membership. Oracle Identity Analytics is also the authoritative source for policy entitlement definitions. (Roles in Oracle Identity Analytics correspond to "groups" in OIM, and policies in Oracle Identity Analytics correspond to "access policies" in OIM.)

- All roles are defined and created in Oracle Identity Analytics. All entitlements for policies and role-to-user relationships are managed from Oracle Identity Analytics.
- Roles managed by Oracle Identity Analytics become read-only in OIM.

Note - Provisioning attribute definitions for Access Policies, which are required to create accounts, is managed in much the same way as the previous Oracle Role Manager(ORM) - OIM integration (by OIM or external process).

Understanding Terminology in Oracle Identity Analytics and Oracle Identity Manager

The following table maps Oracle Identity Analytics terminology to Oracle Identity Manager terminology.

Oracle Identity Analytics Terminology	Oracle Identity Manager Terminology
Resource Type	Resource Type
Resource Type Attributes (NameSpace Attributes)	Provisioning Attributes and Entitlements
Resource	IT Resource
Global Users	Xellerate End Users
Roles	Groups
Policies	Access Policies

To Configure Oracle Identity Analytics and Oracle Identity Manager to Work Together (Deprecated Integration Method)

Before You Begin -

- **At least version 9.1.0.2 BP5 of Oracle Identity Manager and at least version 11gR1 of Oracle Identity Analytics are required.**
- Oracle Identity Manager should be installed and configured.

1. In Oracle Identity Analytics add Oracle Identity Manager as a provisioning server option. ("Sun Identity Manager" and "File" are the default options.)

   See "Step 1: Enable Oracle Identity Manager as a Provisioning Server Option" on page 31
2. Copy the required Oracle Identity Manager API JAR files to Oracle Identity Analytics.

   See "Step 2: Copy the Required `.jar` Files" on page 33

3. In Oracle Identity Analytics, designate Oracle Identity Manager as the provisioning server. Establish a connection by entering authentication details.

   See "Step 3: Designate Oracle Identity Manager as the Provisioning Server" on page 35
4. To send real time changes from Oracle Identity Analytics to Oracle Identity Manager, change the Oracle Identity Analytics configuration files related to workflows.

Step 1: Enable Oracle Identity Manager as a Provisioning Server Option

In the Oracle Identity Analytics user interface, the Administration > Configuration > Provisioning Servers tab displays "file" and "sun" as the available options. To display Oracle Identity Manager as a supported provisioning server, edit `iam-context.xml` in the `RBACX_Home/WEB-INF` folder as follows.

Uncomment the oracle key entry in the iamSolutions property map lines in `iam-context.xml`:

```
<bean id="rbacxIAMService" parent="baseTransactionProxy">
<property name="target">
<bean class="com.vaau.rbacx.iam.service.impl.RbacxIAMServiceImpl" parent="baseServiceSupport">
<property name="iamSolutions">
<map>
<entry key="sun">
<ref local="waveset"/>
</entry>
<!--entry key="ca">
<ref local="eTrust"/>
</entry-->
<!--entry key="ibm">
<ref local="tim"/>
</entry-->
<entry key="oracle">
<ref local="oim"/>
</entry>
<entry key="file">
<ref local="file"/>
</entry>
</map>
</property>
```

and the second change to this file is to uncomment the bean definition:

```
<bean id="oim" class="com.vaau.rbacx.iam.oracle.OIMIAMSolution" parent="abstractIAMSolution">

<property name="metadataManager" ref="metadataManager"/>

<property name = "namespaceMap">
<map>
<!-- This mapping fetches the attributes from
the appropriate object form ( AD User). This
mapping clarifies that, for the "AD Server"
```

```
resource type, attributes are imported from
the "AD User" Object form in OIM -->
<entry key = "AD Server">
<value>AD User</value>
</entry>
</map>
</property>
<property name="resourceFieldMap">
<map>
<!-- This mapping identifies the field that is the
ITResourceLookupField for each resource type.
(Oracle Identity Manager "IT resources" map to
resources in Oracle Identity Analytics.) From the mapping
for the "AD Server" resource type field, we
define that the "UD_ADUSER_AD" column field
corresponds to the ITResource Entry. -->
<entry key="AD Server">
<value>UD_ADUSER_AD</value>
</entry>
</map>
</property>

<property name="accountIdentifierMap">
<map>
<entry key="AD Server">
<value>UD_ADUSER_UID</value>
</entry>
</map>
</property>
<property name = "secPolicyMap">
<map>
<entry key = "RACF Account">
<value>Server,Group</value>
</entry>
</map>
</property>
<property name="maxStaleDays">
<value>${com.vaau.rbacx.iam.oracle.maxStaleDays}</value>
</property>
<property name = "excludeFlag" >
<value>${com.vaau.rbacx.iam.oracle.excludeFlag}</value>
</property>

<property name = 'roleDao'>
<ref bean="roleDao"/>
</property>
<property name = "policyManager">
<ref bean = "policyManager"/>
</property>
<property name="userProperties">
<map>
<entry key = "userName">
<value>Users.User ID</value>
</entry>
<entry key = "firstName">
<value>Users.First Name</value>
</entry>
<entry key = "lastName">
<value>Users.Last Name</value>
```

```
</entry>
<entry key = "middleName">
<value>Users.Middle Name</value>
</entry>
<entry key = "manager">
<value>Users.Manager Login</value>
</entry>
<entry key = "primaryEmail">
<value>Users.Email</value>
</entry>
<entry key = "employeeType">
<value>Users.Role</value>
</entry>
<entry key = "startDate">
<value>Users.Start Date</value>
</entry>
<entry key = "endDate">
<value>Users.End Date</value>
</entry>
<entry key = "createDate">
<value>Users.Provisioned Date</value>
</entry>
</map>
</property>
<property name = "customProperties">
<list>
<value>Users.Email</value>
<value>Organizations.Organization Name</value>
<value>USR_UDF_LOCATION</value>
<value>Users.Deprovisioning Date</value>
<value>Users.Xellerate Type</value>
<value>Users.Identity</value>
<value>Users.Lock User</value>
<value>Users.Disable User</value>
<value>Users.Role</value>
</list>
</property>
</bean>
```

Step 2: Copy the Required `.jar` Files

1. Copy the following Oracle Identity Manager Java API JAR files (located here: `$OIM_HOME/xellerate/lib/.jar`) to the Oracle Identity Analytics `$RBACX_HOME/WEB-INF/lib` folder:
   - `wlXLSecurityProviders.jar`
   - `xlAPI.jar`
   - `xlAuthentication.jar`
   - `xlCache.jar`
   - `xlCrypto.jar`
   - `xlDataObjectBeans.jar`
   - `xlDataObjects.jar`

- `xlLogger.jar`
- `xlScheduler.jar`
- `xlUtils.xls`
- `xLVO.jar`

2. Copy the following Oracle Identity Manager Java API JAR file (located in the `client/ext` folder) to the Oracle Identity Analytics `$RBACX_HOME/WEB-INF/lib` folder:
   - `iam-platform-utils.jar`
3. Copy the following JAR files if you are deploying to a JBoss or WebLogic application server:
   - If deploying to a JBoss application server, copy `jbossall-client.jar`
   - If deploying to a WebLogic application server, copy `oim_design_consolexlclientextwlfullclient.jar`

     Note - The `wlfullclient.jar` is only required if Oracle Identity Analytics and Oracle Identity Manager are on different WebLogic domains. This JAR file allows client applications, such as Oracle Identity Analytics, to communicate with the WebLogic Server over the T3 protocol. If you deploy OIA and OIM to the same WebLogic domain, skip this step, otherwise you may receive an error similar to the following:

     ```
     Caused By: java.lang.LinkageError: loader constraint violation: loader
     (instance of weblogic/utils/classloaders/ChangeAwareClassLoader)
     previously initiated loading for a different type with name
     "javax/xml/namespace/QName"
     ```

     If `wlfullclient.jar` is not present in Oracle Identity Manager, follow these steps to generate it:

     a. Type `cd <WLS-HOME>/server/lib`, where <WLS-HOME> is the base WebLogic installation directory

     b. Type `java -jar wljarbuilder.jar`

     c. Copy the `wlfullclient.jar` file to the `$RBACX_HOME/WEB-INF/lib` folder
4. Copy the following 11g Oracle Identity Manager Java API JAR files to Oracle Identity Analytics:

   a. Copy `$OIM_HOME/server/client/oimclient.jar` to `$OIA-HOME/WEB-INF/lib`

   Note - If this JAR file is not present, you will receive the following exception during

   integrated operations:

   ```
   java.lang.NoClassDefFoundError:oracle/iam/platform/OIMClient at
   Thor.API.tcUtilityFactory.<init>(tcUtilityFactory.java:154) at
   com.vaau.rbacx.iam.oracle.OIMIAMSolution.
   getUtilityFactory(OIMIAMSolution.java:2595) at
   com.vaau.rbacx.iam.oracle.OIMIAMSolution.readUsers(OIMIAMSolution.java)
   ```

   b. Copy the OIM 11g logger JAR file, `xlLogger10g.jar`, to `$OIA-HOME/WEB-INF/lib`

Note - If this JAR file is not present, you will receive the following error during integrated operations:

```
Caused by: java.lang.NoClassDefFoundError:
com/thortech/util/logging/Logger at
Thor.API.tcUtilityFactory.<clinit>(tcUtilityFactory.java:80) at
com.vaau.rbacx.iam.oracle.OIMIAMSolution.
getUtilityFactory(OIMIAMSolution.java:2595) at
com.vaau.rbacx.iam.oracle.OIMIAMSolution.readUsers(OIMIAMSolution.java:770)
at com.vaau.rbacx.iam.service.impl.RbacxIAMServiceImpl.
importUsers(RbacxIAMServiceImpl.java:119)
```

Step 3: Designate Oracle Identity Manager as the Provisioning Server

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Provisioning Servers.
4. Click New Provisioning Server Connection.

   The New Provisioning Server Connection wizard asks you to choose the type of provisioning server connection that you want to create.
5. From the Type of Provisioning Server Connection drop-down menu, select Oracle and click Next.
6. Complete the form:
   - **Server Name** - Type the connection object name.
   - **Xellerate Home** - Type the path to the `config` file in OIM. (example: `C:oraclexellerate`)
   - **Login Config** - Type the path to the authentication configuration (`auth.config`) file. (example: `C:oraclexellerateconfigauth.conf`)
   - **Provider URL** - Type the provider URL. The format for this field is as follows:
     - **WebLogic** -

       `t3://host:7001`
     - **JBoss** -

       `jnp://host:1099` (The default port number in a clustered environment is 1100.)
     - **WebSphere** -

       `corbaloc:iiop:host:2809`
   - **Initial Context Factory** - Enter the name of the environment property for specifying the initial context factory. The default values are as follows:

  - **WebLogic** -

    `weblogic.jndi.WLInitialContextFactory`
  - **JBoss** -

    `org.jnp.interfaces.NamingContextFactory`
  - **WebSphere** -

    `com.ibm.websphere.naming.WsnInitialContextFactory`
- **User Name** - Enter the OIM user name. (example: `xelsysadm`)
- **Password** - Enter the OIM password.

Step 4: Enable Real-Time Updates from Oracle Identity Analytics to Oracle Identity Manager

To send real-time changes from Oracle Identity Analytics to Oracle Identity Manager, change the configuration files related to workflows.

For example, the following code snippet has to be enabled in `role-creation-workflow.xml` during the "Finish" step (step 6):

```
<!--<function name="exportIAMRoleFunction" type="spring">
<arg name="bean.name">exportIAMRoleFunction</arg>
<arg name="iamConnectionName"/>
</function>-->
```

This becomes the following:

```
<function name="exportIAMRoleFunction" type="spring">
<arg name="bean.name">exportIAMRoleFunction</arg>
<arg name="iamConnectionName">OIMConnectionObjectName</arg>
</function>
```

Note — `OIMConnectionObjectName` is the name of the connection object you define in Step 2. Similar changes have to be made for all role related workflows: `role-modification-workflow.xml, role-user-membership-workflow.xml, role-user-membership-activation-workflow.xml`

Populating Oracle Identity Analytics With User Information From Oracle Identity Manager

Refer to the use cases in this section if you have user entitlements in Oracle Identity Manager that you want to use to populate the Oracle Identity Analytics Identity Warehouse. Importing users and roles from Identity Manager into Oracle Identity Analytics should be a one-time event that takes place when first configuring the systems.

Use Case 1: Importing Global Users From Oracle Identity Manager Into Oracle Identity Analytics

The users existing in Oracle Identity Manager (Xellerate End Users) are imported as global users in Oracle Identity Analytics on a scheduled basis. The attributes of the users in OIM are mapped to global user properties in Oracle Identity Analytics by way of a map. Extended attributes in OIM can be imported as custom properties in Oracle Identity Analytics.

The following table contains the default mapping of user attributes between Oracle Identity Analytics and Oracle Identity Manager.

Oracle Identity Analytics User Attribute Name	Oracle Identity Manager (OIM) User Attribute Name
`username`	`Users.UserID`
`firstname`	`Users.First Name`
`lastname`	`Users.Last Name`
`middlename`	`Users.Middle Name`
`manager`	`Users.Manager Login`
`primaryemail`	`Users.Email`
`startdate`	`Users.Start Date`
`enddate`	`Users.End Date`
`createdate`	`Users.Provisioned Date`

To Import Users From Oracle Identity Manager Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Users.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Complete the form by entering the Name and Description of the Job.
7. Choose one of the following tasks:
    - To run the job immediately, select the Run the Job Now option.
    - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
8. Click Finish.

The import users job runs on the scheduled date and time.

9. Verify that the users are imported into Oracle Identity Analytics from Identity Manager by accessing the Users View in Oracle Identity Analytics (choose Identity Warehouse > User).

Use Case 2: Importing Resource Metadata From Oracle Identity Manager Into Oracle Identity Analytics

In the Oracle Identity Analytics integration with Identity Manager, information on resource metadata can be imported from Identity Manager to Oracle Identity Analytics. This eliminates the need to manually recreate resource metadata in Oracle Identity Analytics.

To Import Resource Metadata From Identity Manager Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Resource Metadata.

   The next page will prompt you to choose the resource from the list of available resources for which metadata on attributes needs to be imported.
5. Select the specific resource type.
6. Under Data Selection Source, select the appropriate Connection Name and click Next.
7. Complete the form by entering the Name and Description of the Job.
8. Choose one of the following:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
9. Click Finish to generate the Import Job.

   The import resource metadata job runs on the scheduled date and time.
10. Verify that the resource metadata was properly imported into Oracle Identity Analytics by accessing the Oracle Identity Analytics Resources Types tab (choose Configuration > Resources Types).

Use Case 3: Importing Resources From Identity Manager Into Oracle Identity Analytics

With out-of-the-box integration capabilities, Oracle Identity Analytics can import resources from Oracle Identity Manager to Oracle Identity Analytics. This eliminates the need to manually create the resources in Oracle Identity Analytics. ITResource in OIM corresponds to a resource in Oracle Identity Analytics.

To Import Resources From Identity Manager Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Resources.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Complete the form by typing a name and description for the job.
7. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
8. Click Finish to generate the import job.

   The import resources job runs on the scheduled date and time.
9. Verify that the resources are imported into Oracle Identity Analytics from Identity Manager by accessing the Oracle Identity Analytics Resources tab (choose Identity Warehouse > Resources).

Use Case 4: Importing Roles From Identity Manager Into Oracle Identity Analytics

Groups defined in OIM are imported as Roles within Oracle Identity Analytics. This import also pulls in the relationship between the Group to Access Policy within OIM as Roles-Policy relationship within Oracle Identity Analytics. This requires a successful policy import.

In addition, this step also imports the group-user relationship from OIM and recreates it as a role-user relationship in Oracle Identity Analytics. To establish role-user relationship, ensure that users are imported.

To Import Role From Identity Manager Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Roles.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Complete the form by typing a name and description for the job.
7. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
8. Click Finish to generate the import job.

   The import resources job runs on the scheduled date and time.
9. Verify that the roles are imported into Oracle Identity Analytics from Identity Manager by accessing the Oracle Identity Analytics Roles tab (choose Identity Warehouse > Resources).

Populating Oracle Identity Manager With Roles Information From Oracle Identity Analytics

See the use cases in this section if you have user accounts in Oracle Identity Analytics that you want to use to populate the Identity Manager repository.

Roles defined in Oracle Identity Analytics can be exported to OIM on a scheduled basis, once role definition/management is completed.

This use case will perform the following exports into OIM:

1. Export Oracle Identity Analytics roles to OIM groups.
2. Export the Oracle Identity Analytics policy definition and its entitlements from Oracle Identity Analytics into OIM Access Policies. If the policy does not exist it would create the new policy as Access Policies within OIM.
3. Export the Oracle Identity Analytics Policy-Resource relationship as OIM Access Policy-ITResource relationship.
4. Export the Oracle Identity Analytics Role-Policy relationship as OIM Group-Access Policy relationship.
5. Export the Oracle Identity Analytics Role-User relationship to OIM Group-User relationship.

Note : During initial integration this is done on a scheduled basis. A recommended long-term solution is to update OIM as definitions are changed in Oracle Identity Analytics on a real-time basis.

Use Case 1: Exporting Roles From Oracle Identity Analytics to Identity Manager

Note - Roles in Oracle Identity Analytics correspond to Groups in Identity Manager.

To Export Roles to Identity Manager

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new export job, choose Schedule Job > Export> Export Roles.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Complete the form by entering the Name and Description of the Job.
7. Choose one of the following:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
8. Click Finish to create the Import Job.

   The job runs on the scheduled date and time.
9. Verify that the roles were properly exported to Identity Manager by opening Identity Manager and clicking the User Group -> Manage link on the left pane.

Understanding Closed Loop Compliance

With the integration of Oracle Identity Analytics and Oracle Identity Manager, it is possible to directly revoke roles and entitlements from Oracle Identity Manager if the results of the certification process require it. This integration eliminates the need for manual de-provisioning of access for managed resources. In addition, the manual process of revoking roles and entitlements by leveraging the information stored in the remediation configuration module is also retained. This takes into account non-managed applications.

If certification remediation is enabled, changes are propagated to Oracle Identity Manager either when the certification is complete, or when the certification end-date is reached (depending on configuration). OIM revokes or re-provisions target system accounts based on the revocations and certifications that occurred during the certification process.

To Configure Resources in Oracle Identity Analytics for Remediation

Every resource type in Oracle Identity Analytics can be separately configured for automatic or manual remediation.

1. Log in to Oracle Identity Analytics.
2. Choose Identity Warehouse> Resources.
3. Click the resource for which remediation action needs to be configured, and go to the Remediation tab.
4. Select the Select Provisioning Mode check box.
5. Choose the mode of provisioning desired for the particular resource.
   - **Auto** - Automatically send role/entitlement updates linked with this resource to Oracle Identity Manager.

     Select the appropriate connection name of the provisioning server and save the changes.
   - **Manual** - Use the manual steps for revocation of roles and entitlements using a text editor.

     List the steps to be followed for non-managed system remediation and save the changes.

To Configure Certifications in Oracle Identity Analytics for Remediation

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Identity Certification.
4. Expand the Revoke and Remediation section, and, under the Remediation section, choose one of the following options:
   - **Display Remediation Instructions** - Select to display instructions about how to perform manual remediation of nonmanaged resources.
   - **Perform Closed Loop Remediation on** - Select to specify that the remediation be completed by either the Certification End Date or the Certification Completion Date.

CHAPTER 3

Integrating With Oracle Waveset (Sun Identity Manager)

Overview

Oracle Identity Analytics software and Oracle Waveset software (formerly named Sun Identity Manager) work together seamlessly when integrated using the Service Provisioning Mark-Up Language (SPML). When integrated, Oracle Waveset serves as the automated provisioning and identity synchronization solution, while Oracle Identity Analytics defines the Role-based Access Control (RBAC) framework, the attestation process, and the approach to Segregation Of Duties (SoD) policy enforcement. Rather than assigning individual access entitlements, the RBAC framework allows organizations to assign and unassign roles as a means of controlling user access on various applications.

The Oracle Identity Analytics Identity Warehouse makes it possible for Oracle Identity Analytics to manage users and their identities across various target systems. Before Oracle Identity Analytics features can be utilized, however, the Identity Warehouse of users and their entitlements must be built. If Oracle Waveset is already in use, building the Identity Warehouse is as easy as connecting to Oracle Waveset and importing the user entitlement information that is stored in the Oracle Waveset repository. Roles are then assigned to users, either based on their actual entitlements or business-level attributes. These roles can be exported to Oracle Waveset for user management and provisioning purposes. Additionally, revocations made during the certification campaigns can also be sent from Oracle Identity Analytics to Oracle Waveset so that remediation can take place.

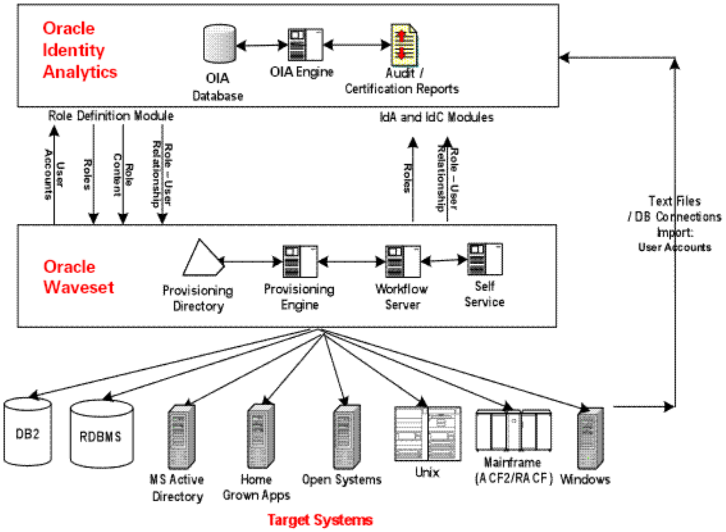
Refer to the *Oracle Identity Analytics 11gR1 User's Guide* for explanations of attributes, attribute categories, resource types, and other concepts.

Oracle Identity Analytics and Oracle Waveset share the following integration points:

- Oracle Waveset *users* are imported into Oracle Identity Analytics
- Oracle Waveset *resources* are imported into Oracle Identity Analytics
- Oracle Waveset *resource metadata* is imported into Oracle Identity Analytics
- Oracle Waveset *user accounts* are imported into Oracle Identity Analytics
- Oracle Identity Analytics *roles* and *role content* are exported to Oracle Waveset

- Closed Loop Compliance

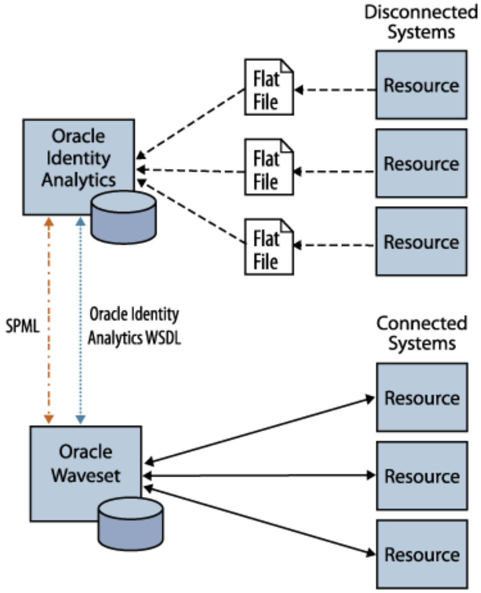
Note – See the *Oracle Identity Analytics Importing* chapter in the *Oracle Identity Analytics 11gR1 Business Administrator's Guide* for more information about the import process.



Integration Architecture

As illustrated in the following figure, Oracle Waveset and Oracle Identity Analytics use SPML and Web Services (WSDL) to communicate. SPML calls are used when Oracle Identity Analytics initiates requests, and Web Services are used when Oracle Waveset initiates the requests.

User and entitlement data can be imported into Oracle Identity Analytics using flat files. In an environment where Oracle Waveset is already deployed, however, (or is in the process of being deployed) Oracle Identity Analytics can connect to Oracle Waveset using SPML to import the user and entitlement data of managed resources. Oracle Identity Analytics can also be used to export roles and user-role membership, and send revocations back to Oracle Waveset.



Integrating Oracle Identity Analytics With Oracle Waveset

This section describes how to configure Oracle Identity Analytics and Oracle Waveset so that the two products can be used together.

To Configure Oracle Identity Analytics and Oracle Waveset to Work Together

Before You Begin -

- **At least version 8.1.1 of Oracle Waveset and at least version 11gR1 of Oracle Identity Analytics are required.**
- Install and configure Oracle Waveset with the Oracle Waveset Gateway.
- In a production environment, deploy Oracle Waveset and Oracle Identity Analytic on separate application servers.
- If you are running Oracle Waveset on the WebLogic application server, install the Metro libraries in the Waveset `WEB-INF/lib` directory. For details, see *Oracle Waveset Installation 8.1.1*, "Installing Waveset on WebLogic," "Step 5: Install the Metro Libraries."

1. In Oracle Waveset, import the SPML Exchange File so that Oracle Waveset can receive (and respond to) SPML requests sent from Oracle Identity Analytics. The SPML Exchange File (`rm_idm_init.xml`) is supplied with Oracle Identity Analytics.

   See "Step 1: To Import the Oracle Waveset SPML Exchange File" on page 46 for details.

2. In Oracle Identity Analytics, create an Oracle Identity Analytics user that Oracle Waveset will use to connect to Oracle Identity Analytics using Web Services. See “Step 2: To Create a Oracle Identity Analytics User That Oracle Waveset Will use to Connect” on page 46 for details.
3. In Oracle Waveset, create an Oracle Waveset user that Oracle Identity Analytics will use to invoke SPML calls to Oracle Waveset. See “Step 3: To Create an Oracle Waveset User That Oracle Identity Analytics Will use to Connect” on page 47 for details.
4. In Oracle Identity Analytics, designate Oracle Waveset as the provisioning server.

   See “Step 4: To Designate Oracle Waveset as the Provisioning Server” on page 47 for details.
5. In Oracle Waveset, add Oracle Identity Analytics Web Services so that Oracle Waveset can send requests to (and receive responses from) Oracle Identity Analytics.

   See “Step 5: To Configure Oracle Waveset to use Oracle Identity Analytics Web Services” on page 48 for details.
6. In Oracle Waveset, configure the User Deferred Task Scanner. This step is required so that real-time Separation of Duties (SoD) processing will work properly.

   See “Step 6: To Configure the User Deferred Task Scanner” on page 50 for details.
7. In Oracle Waveset, configure the User Form so that Oracle Identity Analytics can authenticate over SPML.

   See “Step 7: To Configure the User Form so That Oracle Identity Analytics can Authenticate Over SPML” on page 50 for details.
8. Configure Oracle Identity Analytics for closed loop remediation. For details, see “Understanding Closed Loop Compliance” on page 56.

Step 1: To Import the Oracle Waveset SPML Exchange File

1. Copy the `rm_idm_init.xml` file, which is located in the Oracle Identity Analytics `conf/spml` directory, to the Oracle Waveset server.
2. Log in to Oracle Waveset.
3. Choose Configure > Import Exchange File.
4. Click Browse and navigate to the `rm_idm_init.xml` file.
5. Click Import.

   The exchange file import status is displayed on the Admin Console.
6. Restart the Oracle Waveset application server.

Step 2: To Create a Oracle Identity Analytics User That Oracle Waveset Will use to Connect

1. Log in to Oracle Identity Analytics.

2. Create a user that Oracle Waveset can use to connect to Oracle Identity Analytics using Oracle Identity Analytics Web Services.

   For help creating an Oracle Identity Analytics user, see the *Oracle Identity Analytics 11gR1 Business Administrator's Guide*, "Oracle Identity Analytics Access Control" chapter, *To Create, Update, and Delete an Oracle Identity Analytics User* task.

   a. Assign the user the `SRMAdmin` system role.
   b. Save the user.

Step 3: To Create an Oracle Waveset User That Oracle Identity Analytics Will use to Connect

1. Log in to Oracle Waveset.
2. Create a user that Oracle Identity Analytics can use to invoke SPML calls to Oracle Waveset.

   For help creating an Oracle Waveset user, see the *Oracle Waveset Business Administrator's Guide*, "Administration" chapter, To Create an Administrator task.

   a. If you are using Oracle Waveset 8.1.1, assign the user the "Identity Analytics Admin" admin role, and skip to step c.

      Otherwise, in at least version 8.1.1 of Oracle Waveset, assign the user the following capabilities:

      - Create User
      - Deprovision User
      - Update User
      - Unlink User
      - Unassign User
      - Rename User
      - Enable User
      - Disable User
      - View User
      - Role Administrator

   b. Assign the user control of the `Top` organization.

   c. Assign the user the Empty Form as its User Form.

   d. Save the user.

Step 4: To Designate Oracle Waveset as the Provisioning Server

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Provisioning Servers.
4. Click New Provisioning Server Connection.

The New Provisioning Server Connection wizard asks you to choose the type of provisioning server connection to create.

5. From the Type of Provisioning Server Connection drop-down menu, select Sun and click Next.
6. Complete the form:
   - **Connection Name** - Type a new connection name for Oracle Waveset. This connection name is used during the import process instead of the host name and port.
   - **SPML URL** - Format the SPML URL as follows: `http://`*IdentityManagerApplicationServerName*`:`*PortNumber*`/idm/servlet/rpcrouter2`

     For example: `http://localhost:8080/idm/servlet/rpcrouter2`
   - **Username** - Type a user name that Oracle Identity Analytics will use to connect to Oracle Waveset. You should have created a special Oracle Waveset user account for this purpose in step 3. Do not use the `configurator` account.
   - **Password** - Type the password that Oracle Identity Analytics will use to connect to Oracle Waveset.
   - **Test Connection** - Click to test whether the connection was successfully established between Oracle Waveset and Oracle Identity Analytics. This will help you in troubleshooting connection issues.
   - **Role Consumer** - Select this box to export roles and role content from Oracle Identity Analytics to Oracle Waveset on a real-time basis. Oracle recommends that you select this option.
   - **Role Update Schedule** - Choose to schedule when to send updates back to Oracle Waveset.
     - **Now** - Updates roles in Oracle Waveset as soon as they are updated in Oracle Identity Analytics.
     - **Later**- Schedules the update of roles to take place on a daily, weekly, or monthly basis, or just one time, and schedules the time and date for the update task to start.

Step 5: To Configure Oracle Waveset to use Oracle Identity Analytics Web Services

Oracle Waveset needs to be configured to use Oracle Identity Analytics Web Services. Oracle Waveset uses Oracle Identity Analytics web service calls to both send requests to Oracle Identity Analytics, and receive responses. To configure Oracle Identity Analytics Web Services, use the Oracle Waveset resource wizard.

1. Log in to Oracle Waveset.
2. Choose the Resources tab and verify that the List Resources subtab is selected.
3. Locate the Resource Type Actions drop-down list and select New Resource.

   The New Resource page opens.

4. Select the Oracle Identity Analytics (Sun Role Manager) Web Services resource type from the drop-down list, and click New. (If this resource type is not listed, you need to enable it. See "Managing the Resources List" in the "Roles and Resources" chapter in the *Oracle Waveset Business Administrator's Guide* for details.)

   The Resource Wizard Welcome Page opens.

5. Click Next to begin configuring the Oracle Identity Analytics (Role Manager) Web Services resource.

   The Create Oracle Identity Analytics (Sun Role Manager) Web Services Resource Wizard / Resource Parameters page opens.

6. Complete the form:

   - **Web Service Base URI** - Type the Uniform Resource Identifier (URI) for your Oracle Identity Analytics installation as follows:

     `http://` *server-nameport-number* `/rbacx`

     where *server-name* is the IP address or alias of the server on which Oracle Identity Analytics is running, and *port-number* is the port number of the application server that is listening to Oracle Identity Analytics calls.

   - **User** - Type the user name that Oracle Waveset will use to connect to Oracle Identity Analytics. You should have created a special Oracle Identity Analytics user account for this purpose in step 2. Do not use the `rbacxadmin` account.

   - **Password** - Type the password that Oracle Waveset will use to connect to Oracle Identity Analytics.

   - **Oracle Identity Analytics Version** - Type the version number of Oracle Identity Analytics that Oracle Waveset is connecting to.

   - **Is SRM Configured** - Type `true` to enable Oracle Waveset to use Oracle Identity Analytics Web Services.

   - **Test Configuration** - Click to test the connection to Oracle Identity Analytics Web Services.

     Note - Upon completing the wizard, additional form fields are unlocked. These fields include the following:

   - **Process Check Policy Results Rule** - Value should be `Sun Role Manager:Process Policy Result`

   - **Check Policy Compliance Violation Form** - Value should be `Sun Role Manager Compliance Violation Form`

   - **Check Policy Status Rule** - Value should be `Sun Role Manager:Risk Analysis Status`

   - **Compliance Violation Owners Rule** - Value should be `Sun Role Manager:Compliance Violation Owners`

7. Click Next.

The Create Oracle Identity Analytics (Sun Role Manager) Web Services Resource Wizard / Account Attributes page opens.

8. Verify that the account attribute mappings on this page are correct and click Next.

   The Create Oracle Identity Analytics (Sun Role Manager) Web Services Resource Wizard / Identity Template page opens.

9. Verify that the attribute value in the Identity Template box is correct and click Save.

Step 6: To Configure the User Deferred Task Scanner

The User Deferred Task Scanner in Oracle Waveset needs to be configured for a delay of one minute so that SoD processing will work properly. The scanner picks up SoD information after it has been retrieved from Oracle Identity Analytics using Oracle Identity Analytics (Sun Role Manager) web services.

1. Log in to Oracle Waveset.
2. Choose Server Tasks > Manage Schedule.
3. Click User Deferred Task Scanner to edit the task.

   The Edit Task Schedule page opens.
4. Change the value in the Repeat Every box to a value of `1 Minutes`.
5. Click Save.

Step 7: To Configure the User Form so That Oracle Identity Analytics can Authenticate Over SPML

Within Identity Manger, the User Form of the user that Oracle Identity Analytics authenticates as over SPML needs to be set to "Empty Form."

1. Log in to Oracle Waveset.
2. Choose the Accounts tab and verify that the List Accounts subtab is selected.
3. Click the user that you created in “Step 3: To Create an Oracle Waveset User That Oracle Identity Analytics Will use to Connect” on page 47.

   The Edit User page opens.
4. Click the Security tab.
5. From the User Form drop-down box, select Empty Form.
6. Click Save.

Oracle Identity Analytics and Oracle Waveset are now configured to work together. To configure closed loop remediation, see “Understanding Closed Loop Compliance” on page 56.

Populating Oracle Identity Analytics With User Information From Oracle Waveset

Refer to the use cases in this section if you have user entitlements in Oracle Waveset that you want to use to populate the Oracle Identity Analytics Identity Warehouse. Importing users and roles from Oracle Waveset into Oracle Identity Analytics should be a one-time event that takes place when first configuring the systems.

Use Case 1: Importing Global Users From Oracle Waveset Into Oracle Identity Analytics

Oracle Waveset saves information about users who are auto-provisioned. These users are imported into Oracle Identity Analytics as global users before their accounts are pulled in.

To Import Users From Oracle Waveset Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Users.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Complete the form by entering the Name and Description of the Job.
7. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
8. Click Finish.

   The import users job runs on the scheduled date and time.
9. Verify that the users are imported into Oracle Identity Analytics from Oracle Waveset by accessing the Users View in Oracle Identity Analytics (choose Identity Warehouse > User).

Use Case 2: Importing Resource Metadata From Oracle Waveset Into Oracle Identity Analytics

A *resource type* in Oracle Waveset is a type of target system, whereas a *resource* is an instance of a resource type. For example, consider the case of four different Windows NT systems hosting four different sets of users. In this scenario, 'Windows NT' is the resource type, whereas the four individual system names are resources of type 'Windows NT.'

In the Oracle Identity Analytics integration with Oracle Waveset, information on resource metadata can be imported from Oracle Waveset to Oracle Identity Analytics. This eliminates the need to manually recreate resource metadata in Oracle Identity Analytics.

To Import Resource Metadata From Oracle Waveset Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Resource Metadata.

   The next page will prompt you to choose the resource from the list of available resources for which metadata on attributes needs to be imported.
5. Select the specific resource type.
6. Under Data Selection Source, select the appropriate Connection Name and click Next.
7. Complete the form by entering the Name and Description of the Job.
8. Choose one of the following:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
9. Click Finish to generate the Import Job.

   The import resource metadata job runs on the scheduled date and time.
10. Verify that the resource metadata was properly imported into Oracle Identity Analytics by accessing the Oracle Identity Analytics Resources Types tab (choose Configuration > Resources Types).

Note - Seven resource types in Oracle Waveset are treated differently by Oracle Identity Analytics. They are the following:

1. Simulated
2. Scripted JDBC
3. Database Table
4. External
5. Scripted Gateway
6. Scripted Host
7. Shell Script

Each resource within the above resource type is created as a resource_type within Oracle Identity Analytics. The naming convention is "ResourceName__ResourceTypeName". This is because each resource is likely to have its own resource type metadata rather than a common metadata format.

Use Case 3: Importing Resources From Oracle Waveset Into Oracle Identity Analytics

With out-of-the-box integration capabilities, Oracle Identity Analytics can import resources from Oracle Waveset to Oracle Identity Analytics. This eliminates the need to manually create the resources in Oracle Identity Analytics.

To Import Resources From Oracle Waveset Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Resources.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Complete the form by typing a name and description for the job.
7. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
8. Click Finish to generate the import job.

   The import resources job runs on the scheduled date and time.
9. Verify that the resources are imported into Oracle Identity Analytics from Oracle Waveset by accessing the Oracle Identity Analytics Resources tab (choose Identity Warehouse > Resources).

Use Case 4: Importing User Accounts From Oracle Waveset Into Oracle Identity Analytics

After global users are imported, you can import accounts (user entitlements) into Oracle Identity Analytics for different resource types. Before importing user accounts, make sure that the resource types and attributes are correctly configured in Oracle Identity Analytics. For more information, see *Resource Types Configuration* in the *Oracle Identity Analytics 11gR1 Business Administrator's Guide*, Oracle Identity Analytics Configuration chapter.

To Import Accounts From Oracle Waveset Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Accounts, and then click Next.
5. From the list of available resources for which user accounts can be imported, select the resource and the specific resource type.
6. Under Data Selection Source, select the appropriate Connection Name and click Next.
7. Complete the form by entering the Name and Description of the Job.
8. Choose one of the following:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
9. Click Finish to create the Import Job.

   The job runs on the scheduled date and time.
10. Verify that the accounts imported into Oracle Identity Analytics match the corresponding resource type accounts in Oracle Waveset.

Use Case 5: Importing Roles From Oracle Waveset Into Oracle Identity Analytics

Note - This should be done only as a one time effort for initial Roles population. It is recommended that Oracle Identity Analytics be kept as the Authoritative Source for roles and the roles would be overwritten if they were imported from Oracle Waveset on an ongoing basis.

To Import Role From Oracle Waveset Into Oracle Identity Analytics

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.
4. To start a new import job, choose Schedule Job > Import > Import Roles.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Complete the form by typing a name and description for the job.
7. Choose one of the following tasks:
   - To run the job immediately, select the Run the Job Now option.

- To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.

8. Click Finish to generate the import job.

   The import resources job runs on the scheduled date and time.
9. Verify that the roles are imported into Oracle Identity Analytics from Oracle Waveset by accessing the Oracle Identity Analytics Roles tab (choose Identity Warehouse > Resources).

Populating Oracle Waveset With Roles Information From Oracle Identity Analytics

See the use cases in this section if you have user accounts in Oracle Identity Analytics that you want to use to populate the Oracle Waveset repository.

Note - Exporting roles from Oracle Identity Analytics to Oracle Waveset should be a one-time event that takes place during configuration. To export roles to Oracle Waveset, be sure that the Role Consumer box is selected in the Sun (Oracle Waveset) Provisioning Server settings.

Oracle Identity Analytics can create roles based on either existing entitlements or business attributes (client requirements). Policy formation and role-policy association can be performed during role creation. In addition, the role-user association can also be established.

Oracle Waveset does not have the concept of policies. The roles in Oracle Identity Analytics are mapped to Business Roles in Oracle Waveset, whereas the policies in Oracle Identity Analytics are mapped to IT Roles in Oracle Waveset. As policies are directly assigned to resources in Oracle Identity Analytics, similarly IT Roles are directly assigned to resources in Oracle Waveset. Thus, the one-to-many relationship between role and policies is carried forward from Oracle Identity Analytics to Oracle Waveset by way of the one-to-many relationship between Business Roles and IT Roles. This allows for more efficient grouping of entitlements and easier management of user access. Thus, along with roles, policies also need to be exported from Oracle Identity Analytics to Oracle Waveset.

Use Case 1: Exporting Roles From Oracle Identity Analytics to Oracle Waveset

Note - Roles in Oracle Identity Analytics correspond to Business Roles in Oracle Waveset.

To Export Roles to Oracle Waveset

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Import/Export.

4. To start a new export job, choose Schedule Job > Export > Export Roles.
5. Under Data Selection Source, select the appropriate Connection Name and click Next.
6. Complete the form by entering the Name and Description of the Job.
7. Choose one of the following:
   - To run the job immediately, select the Run the Job Now option.
   - To schedule the job for later, clear the Run the Job Now option and enter the details of the scheduled job.
8. Click Finish to create the Import Job.

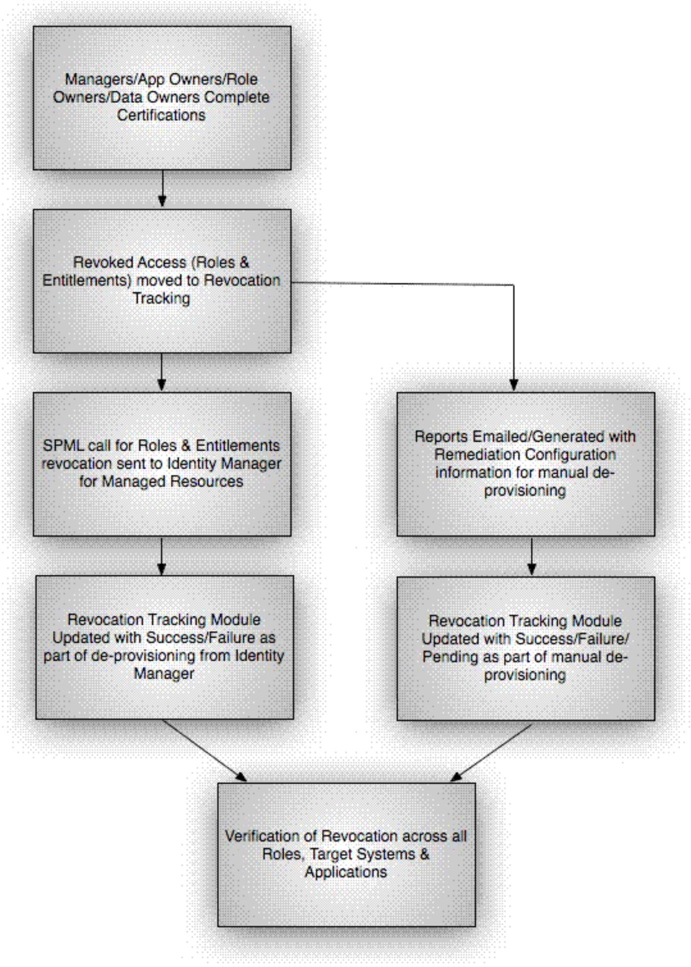
   The job runs on the scheduled date and time.
9. Verify that the roles were properly exported to Oracle Waveset by opening Oracle Waveset and clicking the 'Business Role' Roles tab.

Note - Policies (roles content) are exported as part of roles export.

Understanding Closed Loop Compliance

With the integration of Oracle Identity Analytics and Oracle Waveset, it is possible to directly revoke roles and entitlements from Oracle Waveset if the results of the certification process require it. This integration eliminates the need for manual de-provisioning of access for managed resources. In addition, the manual process of revoking roles and entitlements by leveraging the information stored in the remediation configuration module is also retained. This takes into account nonmanaged applications.

The following closed loop remediation diagram illustrates this process. (Note: In the diagram, Oracle Waveset is referred to by its previous name, Sun Identity Manager.)


Managers/App Owners/Role Owners/Data Owners Complete Certifications
Revoked Access (Roles & Entitlements) moved to Revocation Tracking
SPML call for Roles & Entitlements revocation sent to Identity Manager for Managed Resources
Reports Emailed/Generated with Remediation Configuration information for manual de-provisioning
Revocation Tracking Module Updated with Success/Failure as part of de-provisioning from Identity Manager
Revocation Tracking Module Updated with Success/Failure/ Pending as part of manual de-provisioning
Verification of Revocation across all Roles, Target Systems & Applications

To Configure Resources in Oracle Identity Analytics for Remediation

Every resource type in Oracle Identity Analytics can be separately configured for automatic or manual remediation.

1. Log in to Oracle Identity Analytics.
2. Choose Identity Warehouse > Resources.
3. Click the resource for which remediation action needs to be configured, and go to the Remediation tab.

4. Select the Select Provisioning Mode check box.
5. Choose the mode of provisioning desired for the particular resource.
   - **Auto** - Automatically send role/entitlement updates linked with this resource to Oracle Waveset.

     Select the appropriate connection name of the provisioning server and save the changes.
   - **Manual** - Use the manual steps for revocation of roles and entitlements using a text editor.

     List the steps to be followed for non-managed system remediation and save the changes.

To Configure Certifications in Oracle Identity Analytics for Remediation

1. Log in to Oracle Identity Analytics.
2. Choose Administration > Configuration.
3. Click Identity Certification.
4. Expand the Revoke and Remediation section and, under the Remediation section, choose one of the following options:
   - **Display Remediation Instructions** - Select to display instructions about how to perform manual remediation of nonmanaged resources.
   - **Perform Closed Loop Remediation on** - Select to specify that the remediation be completed by either the Certification End Date or the Certification Completion Date.

Oracle Waveset Sample Workflows

Sample Oracle Waveset workflows are available to facilitate the integration of Oracle Waveset with Oracle Identity Analytics. The most recent sample workflows are located in Appendix A, Appendix A, "Oracle Waveset Sample Workflows," or you can use the sample workflows included with Oracle Waveset 8.1.1–Patch 1 (located in the `sample/wfrolemanager.xml` file).

Note - Do not use the sample workflows included with Oracle Waveset 8.1.1 because they are no longer current.

The following Oracle Waveset sample workflows are available.

Workflow Name	Description
Check SRM Integration	Invokes workflow services to determine if Oracle Identity Analytics (Sun Role Manager) integration has been configured. Returns a Boolean value in the `isSRMIntegrated` variable.

Workflow Name	Description
Merge SRM Role Assignments	If Oracle Identity Analytics is integrated and the `UserView` option `getRuleDrivenRoleManagerRoles` is set to `true`, this process will retrieve the list of roles to be automatically assigned by OIA configured rules. This list of roles will be merged with the Waveset-assigned roles into the `UserView`.
Create SRM User	If Oracle Identity Analytics is integrated, this process invokes the create OIA user action based on `UserView` attributes.
Update SRM User	If Oracle Identity Analytics is integrated, this process invokes the update OIA user action based on `UserView` attributes.
Rename SRM User	If Oracle Identity Analytics is integrated, this process invokes the rename OIA user action.
Delete SRM User	If Oracle Identity Analytics is integrated, this process invokes the delete OIA user action.
Disable SRM User	If Oracle Identity Analytics is integrated, this process invokes a disable OIA user action.
Enable SRM User	If Oracle Identity Analytics is integrated, this process invokes an enable OIA user action.
Create SRM User Reconcile Response Workflow	If Oracle Identity Analytics is integrated, this per-account workflow invokes the creation of OIA users while processing unmatched accounts during reconciliation.

Oracle Identity Analytics Web Services

With an out-of-the-box integration, web services from both Oracle Waveset and Oracle Identity Analytics can be used as needed. For information about Oracle Identity Analytics web services, see the *Oracle Identity Analytics 11gR1 API Guide*.

Troubleshooting

The information in this section briefly describes how to approach troubleshooting a Oracle Identity Analytics and Oracle Waveset integration.

System Logs

Application logs are generated and stored in the application deployment folder in `rbacx.log`. The log captures various details such as import/export information, ETL processing, and any exceptions that can arise while running the application. There are different levels of logging in the `rbacx.log` file, and these can be adjusted and modified as needed. The properties file that is used to alter the logging level is found under the `$RBACX_HOME\WEB-INF\` folder, and the file name is `log4j.properties`.

There are three levels of logging that are commonly used by the system integrators: WARN, INFO, and DEBUG.

To change logging levels, open log4j.properties in a text editor and modify the line under the \#Role Manager IAM logging section as follows:

```
log4j.logger.com.vaau.Role Manager.iam=DEBUG
```

Other parameters to be aware of are Security logging and IAM logging. These logs report Security and entitlement data exceptions.

For more information about logging, see the *Oracle Identity Analytics 11gR1 System Administrator's Guide*.

CHAPTER 4

Integrating With Other Provisioning Servers

Integrating With Other Provisioning Servers Introduction

In addition to Chapter 1, "Integrating With Oracle Identity Manager, Preferred Method," and Chapter 3, "Integrating With Oracle Waveset (Sun Identity Manager)," Oracle Identity Analytics has the ability to integrate with the following provisioning servers:

- IBM Tivoli Identity Manager
- CA eTrust Identity Access Management

In the user interface, the Administration > Configuration > Provisioning Servers tab displays 'file' and 'sun' as the available options. To display other supported provisioning servers, edit `iam-context.xml` in the `RBACX_Home/WEB-INF` folder.

CHAPTER 5

Authenticating With LDAP

To configure Oracle Identity Analytics for use with the Lightweight Directory Access Protocol (LDAP), you need to edit the `ldap.properties` file.

Before you begin the installation, you will need to know the following:

1. The login and password details for the database server system administrator account
2. Sufficient system privileges, such as `sudo`, to modify file and folder permissions
3. The LDAP *base_dn*. The *base_dn* is the starting root under which the users are present.

   For example:

   ```
   DC=vaau,DC=corp,DC=net
   ```
4. A service account information (username and password) with read-only privileges for the LDAP if the LDAP does not allow anonymous connections for search. AD integration by default allows anonymous access.

Configuring `ldap.properties`

The `ldap.properties` file is located in the `$RBACX_HOME/conf` folder.

To Configure the LDAP URL and BASE_DN

The LDAP URL specifies how to connect to and search the LDAP server. The URL takes the following form:

`ldap://`*hostname*`:`*portnumber*`/` or

`ldaps://`*hostname*`:`*portnumber*`/` - **(For SSL)**

where:

- *hostname* is the name (or IP address in dotted format) of the LDAP server.
- *portnumber* is the port number of the LDAP server (for example, `49153`). The default standard LDAP port is `389`.

The LDAP URL is identified by the `ldapAuthentication.url` field in the `ldap.properties` file. Multiple URLs can be specified by using a semicolon (`;`) as a delimiter.

For example: `ldap://vaau1123:389/;ldap://vaau1398:389/`

The Base_DN is identified by the `ldapAuthentication.rootContext` field in the `ldap.properties` file. If Multiple URLs are used, for each URL a corresponding Base_DN must be defined using a semicolon (`;`) as a delimiter.

For example: `DC=vaau,DC=corp,DC=net{}; {}DC=vaau,DC=corp,DC=net`

To Configure the Domain Name

If Active Directory (AD) is being utilized as an LDAP server, the full Windows name of an AD object is in the form `NetBIOSDomain\sAMAccountName`.

For example: `Vaau\rbacxadmin`

The domain name can be configured such that users do not need to type the `domainName\username sAMAccountName` prefix. To do this, uncomment the `ldapAuthentication.userContextPrefix` field in `ldap.properties` and set it to the correct domain name.

For example:

If users log in using `Vaau` as the domain, and `Vaau\rbacxadmin` as the user's log in, then uncomment the `ldapAuthentication.userContext` field and edit it as follows: `ldapAuthentication.userContext=Vaau//`

If Multiple URLs are used, a userContext must be specified for each of the URL separated by a semicolon (;).`ldapAuthentication.userContext=Vaau//;Vaau//`

Note 1 - For AD

1. Oracle Identity Analytics users need to be created using the format *domain/username* in the Oracle Identity Analytics database. In order to create the user as just *username* in the RM database the following parameter needs to be set to false. Uncomment the `ldapAuthentication.keepContextPrefix` field in `ldap.properties` and set it to `false`.
2. A double slash at the end of the domain name is mandatory. The above settings are specific to AD. For all other LDAP only the following have to be configured correctly.

Note 2 - For Non-AD

1. Uncomment the `ldapAuthentication.keepContextPrefix` field in `ldap.properties` and set it to `false`.
2. Uncomment the `ldapAuthentication.isAD` field in `ldap.properties` and set it to `false`.
3. If LDAP does not allow anonymous login, the following parameters should be uncommented and set with the values received from the LDAP administrator for the service account.

   ```
   ldapAuthentication.securityPrincipal=CN=User,DC=vaau,DC=corp,DC=net
   ```

   ```
   ldapAuthentication.securityCredential=password
   ```

If Multiple URLs are used, above parameters must be specified for each of the URL separated by a semicolon (;).

To Configure User Context Search

The `ldapAuthentication.userContext` field can be used as a filter to look for authorized users.

In the event of AD acting as the directory server, `ldapAuthentication.userContext` is specified as {0}. If multiple AD URLs are utilized, then a filter has to be specified for each URL separated by a semicolon (;).

For example:

```
ldapAuthentication.url=ldap://vaau1123:389/DC=vaau,DC=corp,DC=net;
ldap://vaau1398:389/DC=vaau,DC=corp,DC=net
```

Next, set `ldapAuthentication.userContext={0};{0}`.

To Configure the User Account Search Key

The `ldapAuthentication.userAccountSearchKey` field specifies the user account search key. This field is used to retrieve the user name and is set to the default value `sAMAccountName`.

If `sAMAccountName` is not utilized (AD), then set `ldapAuthentication.userAccountSearchKey` to the starting value of user object usually "`cn`" (Sometimes this value could be "`uid`").

To Configure a First Name Search Key

`ldapAuthentication.firstNameSearchKey` specifies the first name search key and is set to `givenName` by default.

To Configure a Last Name Search Key

ldapAuthentication.lastNameSearchKey specifies the last name search key and is set to sn by default.

Note:

If Multiple URLs are used, above parameters must be specified for each of the URL separated by a semicolon (;).

Example of ldap.properties

The following is a snippet of an ldap.properties file that has AD configured as the intended authenticating server.

```
ldapAuthentication.enabled=true
ldapAuthentication.tryNextProviderIfNoAuthenticated=false
ldapAuthentication.stopIfCommunicationError=true

ldapAuthentication.url=ldap\://localhost:389/DC=vaau,DC=corp,DC=net
ldapAuthentication.userContextPrefix=Vaau
ldapAuthentication.keepContextPrefix=false - // This is optional
ldapAuthentication.userContext={0}

ldapAuthentication.userAccountSearchKey=sAMAccountName
ldapAuthentication.firstNameSearchKey=givenName
ldapAuthentication.lastNameSearchKey=sn
```

◆ ◆ ◆ CHAPTER 6

Integrating With Intellitactics Security Manager

Integrating With Intellitactics Security Manager Introduction

Intellitactics Security Manager (ISM) is a Security and Information Event Management (SIEM) solution that enables real-time event monitoring and reporting, and makes it possible to quickly investigate alerts and incidents. By integrating Oracle Identity Analytics with Intellitactics, a certifier can view important user activity information during the certification process.

The inclusion of event and alert information in the certification process allows the certifier to make better, more informed decisions about whether a user should have certain system or application privileges. This information provides Oracle Identity Analytics with a view of *what the user did* (actions), which complements and expands on the default Oracle Identity Analytics view of *what the user is allowed to do* (permissions). This correlation is critical for many compliance standards, including Sarbanes-Oxley (SOX).

CHAPTER 7

Configuring Oracle Identity Analytics For Web Access Control

This chapter describes how to authenticate with Oracle Identity Analytics using Web Access Components.

Overview

Oracle Identity Analytics can be integrated with Web Access Control solutions such as Sun Access Manager, CA's eTrust SiteMinder, Novell's ICHAIN, and so on. This enables Oracle Identity Analytics to follow enterprise standards for web application security.

Configuring Oracle Identity Analytics For Web Access Control

The following two configuration changes need to be made in Oracle Identity Analytics:

1. Setting up the correct HTTP header variable name in `security-context.xml`
2. Setting up the logout URL

To Set Up the `http` Reader

Web Access Control Solutions send user information as part of the `http` header variable. This header variable, which is the user name, holds a unique identity for the user being authenticated. This user name should be the same as the Oracle Identity Analytics user.

As shown in the following snippet from the `security-context.xml` configuration file (under the `WEB-INF` folder in Oracle Identity Analytics), Oracle Identity Analytics is configured to use the value of the `"sm-user"` `http` header variable to authorize a user. Change the property of `preAuthEnabled` to *true* and also change `sm-user` for `preAuthUsernameHeaderKey` and `preAuthPasswordHeaderKey` to the header variable sent by the Web Access Control Solution.

```
   <bean id="preAuthAwareAuthenticationProcessingFilter"
          class="com.vaau.commons.springframework.security.filter.PreAuthAwareAuthenticationProcessi /
ngFilter">
        <property name="authenticationManager">
            <ref bean="authenticationManager"/>
        </property>
        <property name="authenticationFailureUrl" value="/welcome.action?login_error=true"/>
        <property name="defaultTargetUrl" value="/secure/checkExpiredCredentials.action"/>
        <property name="filterProcessesUrl" value="/j_acegi_security_check"/>
        <property name="formUsernameParameterKey" value="j_username"/>
        <property name="formPasswordParameterKey" value="j_password"/>
        <property name="preAuthEnabled" value="true"/>
        <property name="preAuthUsernameHeaderKey" value="sm-user"/>
        <property name="preAuthPasswordHeaderKey" value="sm-user"/>
        <!--SM_USER -->
        <property name="exceptionMappings">
            <props>
                <prop key="org.springframework.security.BadCredentialsException"> /
/welcome.action?login_error=true</prop>
                <prop key="org.springframework.security.CredentialsExpiredException"> /
/passwordExpired.action</prop>
            </props>
        </property>
    </bean>
```

To Set Up the Logout URL

For a user to completely log out from the session, the Oracle Identity Analytics default logout URL needs to be modified with the logout URL for the Web Access Control Solution.

To configure the logout URL in Oracle Identity Analytics, change the following entry in the header.jspf file under the WEB-INF/jspf folder.

Current information in line 111-122 in the header.jspf file:

```
<tr>
   <td height="22">
       <div align="center" style="font-size:10px;">
               <a href="<%=ctx%>/secure/home/home.action" class="hoverUnderline"
    style="color:#000000">Home</a>
               <a href="<%=ctx%>/logout.action" class="hoverUnderline"
    style="color:#000000">Logout</a>
               <a href="<%=ctx%>/docs/userguide/index.html" target="_blank"
 class="hoverUnderline" style="color:#000000">Help</a>
      </div>
   </td>
</tr>
```

Line 111-122 in the header.jspf file after the modification:

```
<tr>
   <td height="22">
       <div align="center" style="font-size:10px;">
<a href="<%=ctx%>/secure/home/home.action" class="hoverUnderline"
```

```
                       style="color:#000000">Home</a> |
                    <a href="www.vaau.com/logout.jsp" class="hoverUnderline"
                       style="color:#000000">Logout</a> |
                    <a href="<%=ctx%>/docs/userguide/index.html" target="_blank"
class="hoverUnderline" style="color:#000000">Help</a>
       </div>
   </td>
</tr>
```

To Access Oracle Identity Analytics When Using Web Access Control

End-users should use the following URL to access Oracle Identity Analytics:

`http://`*OiaHost*:*Port*`/rbacx/j_acegi_security_check`

Note - If the SSO solution allows for setting up a specific redirect URL for each application, then the SSO solution should be configured to redirect to the URL provided above.

Because this URL is protected by the SSO solution, the end-user is redirected to the SSO login screen, and, once successfully authenticated, re-directed to the URL provided. At this point, Oracle Identity Analytics can verify the HTTP header and allow the end-user to access the application.

CHAPTER 8

Customizing The Oracle Identity Analytics User Interface

This chapter describes how to customize the Oracle Identity Analytics user interface (UI).

Oracle Identity Analytics features a rich AJAX Web 2.0 user interface for an enhanced, user-friendly experience. Menu items and logos can be customized so that your organization can adhere to its internal style guidelines.

Before You Begin

To customize the user interface, you need the following access privileges:

- Access to the Oracle Identity Analytics application server with rights to modify and add files to the Oracle Identity Analytics deployed war folder
- Administrative credentials to log in to the Oracle Identity Analytics application

Configuring Logos

The Oracle Identity Analytics home screen displays the default logo.

If Oracle Identity Analytics is hosted on the Apache Tomcat application server, the directory where the .war file is expanded is usually set to the following location:

On UNIX

```
/usr/local/Vaau/rbacx-4.0/tomcat55/webapps/rbacx/
```

On Windows

```
C:\Program Files\Vaau\RBACx2008\tomcat55\webapps\rbacx
```

This path is referred to as $RBACX_WAR in this chapter.

Note - If you are using an application server other than Tomcat, contact a system administrator to determine the location of the deployed Oracle Identity Analytics WAR file.

To Configure a Custom Logo

1. Open the `$RBACX_WAR/images` directory and replace the `logo.gif` file with your company logo.

   Ensure that the company logo follows the same naming convention, which is `logo.gif`.
2. Open Oracle Identity Analytics to view the new logo.

   The new logo is displayed throughout the application.
3. If the new logo is not displayed immediately, restart the application server.

Configuring Labels

All labels in Oracle Identity Analytics can be modified or renamed as desired.

To make changes to labels, it is important to understand the structure of two files: `rbacxmessages.properties` and `rbacxaudit-messages.properties`. These two files contain the dynamic links to configure labels for the Oracle Identity Analytics user interface.

These files are located in `$RBACX_WAR/WEB-INF/classes`.

- The `rbacxmessages.properties` file contains labels that are separated by modules such as Identity Warehouse, Role Management, Identity Certification, and so on.
- The `rbacxaudit-messages.properties` file allows modifications to labels only within the Identity Audit module.

The following procedures describe how to modify the labels of various modules and menu items.

To Modify Menu Labels

1. Open the `rbacxmessages.properties` file located in `$RBACX_WAR/WEB-INF/classes`.
2. Scroll down to the `# Menu` section of the file.
3. As needed, modify the existing menu definitions, which are located to the right of the '`=`' sign, and save the file.
4. Restart the Oracle Identity Analytics application in a new browser window to view your changes.

An excerpt from the `rbacxmessages.properties` file follows:

```
# Menus
menu.welcome=<span>Welcome</span>
menu.register=Register
menu.info=My info
menu.administration=<span Title='Administration'>Administration
  <img src="/rbacx/images/arrow.gif"/></span>
menu.monitoring=Monitoring
menu.logout=Log out
menu.tools=Tools
menu.reports=Reports
menu.dashboard=<span Title='My Reports'> My Reports</span>
menu.security=Security
menu.help=Help
menu.certifications=<span Title='Identity Certification'>Identity
  Certification</span>
menu.provisioning=Access Control
menu.audit=<span Title='Identity Audit'>Identity Audit</span> menu.home=Home
menu.configuration=Configuration
menu.settings=<span Title='My Settings'>My Settings</span>
menu.requests=<span Title='My Requests'>My Requests</span>
menu.system=System
menu.identityWarehouse=<span Title='Identity Warehouse'>Identity Warehouse</span>
menu.users= Users
menu.roles= Roles
menu.businessUnit=Business Unit
menu.reporting=<span Title='Reports'>Reports</span>
menu.roleManagement=<span Title='Role Management'>Role Management</span>
menu.roleEngineering=<span Title='Role Engineering'>Role Engineering</span>
menu.roleEntitlementDiscovery=Role Entitlement Discovery
menu.roleConsolidation=Role Consolidation
menu.myRequest=<span Title='My Request'>My Request</span>
menu.rme=<span Title='Role Engineering'>Role Engineering</span>
```

To Modify User Labels

1. Open the `rbacxmessages.properties` file located in `$RBACX_WAR/WEB-INF/classes`.
2. Scroll down to the `# Identity Warehouse` section of the file.
3. As needed, modify any of the existing user labels, which are located to the right of the "=" sign, and save the file.
4. Restart the Oracle Identity Analytics application in a new browser window to view your changes.

An excerpt from the `rbacxmessages.properties` file follows:

```
################Identity Warehouse ##########################

user.username= User Name
user.employeeid= Employee Id
user.employeetype= Employee Type
user.firstname= First Name
user.middlename= Middle Name
user.lastname= Last Name
user.allNameRequired=All name’s required. user.fullname= Full Name
user.title= Title
user.officename= Office Name
user.street= Street
user.city= City
user.state= State/Province
user.zip= Zip/Postal Code
user.country= Country/Region
user.phone= Phone
user.extension= Extension
user.mobile= Mobile
user.fax= Fax
user.filter.pagesize=Page Size user.pager= Pager
user.pemail= Primary Email
user.semail= Secondary Email
user.comments= Comments
user.suspension= Suspension
user.gustatus= Global User Status user.startdate= Start Date
user.enddate= End Date
user.servicedesk= Service Desk user.status= Status
user.servicedeskticket= Service Desk Ticket
user.serverDeskTicket=Server Desk Ticket
user.customProperty1=Custom Property 1
user.customProperty2=Custom Property 2
user.customProperty3=Custom Property 3
user.customProperty4=Custom Property 4
user.customProperty5=Custom Property 5
user.customProperty6=Custom Property 6
user.customProperty7=Custom Property 7
user.customProperty8=Custom Property 8
user.customProperty9=Custom Property 9
user.customProperty10=Custom Property 10
user.customProperty11=Custom Property 11
user.customProperty12=Custom Property 12
user.customProperty13=Custom Property 13
user.customProperty14=Custom Property 14
user.customProperty15=Custom Property 15
user.customProperty16=Custom Property 16
user.customProperty17=Custom Property 17
user.customProperty18=Custom Property 18
user.customProperty19=Custom Property 19
user.customProperty20=Custom Property 20
user.addBusinessUnit=Add Business Unit
user.address=Address
```

Configuring Error Messages

Configuring error messages in Oracle Identity Analytics is similar to configuring labels.

The rbacxmessages.properties and rbacxaudit-messages.properties files contain the dynamic links to

configure error messages for the Oracle Identity Analytics user interface.

- The rbacxmessages.properties file contains error messages that are separated by modules (such as Identity Warehouse, Role Management, Identity Certification, and so on).

- The rbacxaudit-messages.properties file allows modifications to error messages only within the Oracle Identity Analytics Identity Audit Module.

These files are located at $RBACX_WAR/WEB-INF/classes.

The following procedures describe how to modify the error messages generated from various Oracle Identity Analytics modules.

To Modify My Requests Error Messages

1. Open the rbacxmessages.properties file located in $RBACX_WAR/WEB-INF/classes.
2. Scroll down to the #My Requests section of the file.
3. As needed, modify the existing error message labels, which are located to the right side of the '=' sign, and save the file.
4. Restart the Oracle Identity Analytics application in a new browser window to view your changes.

An excerpt from the rbacxmessages.properties file follows:

```
####################################################
#    My Requests
####################################################
request.error.selectRequest=Please choose a request first!
request.error.approveFailed=Unable to approve the request!
request.error.rejectFailed=Unable to reject the request!
```

To Modify Identity Certification Error Messages

1. Open the rbacxmessages.properties file located in $RBACX_WAR/WEB-INF/classes.
2. Scroll to the #Identity Certification section of the file.
3. As needed, modify the existing error message labels, which are located to the right of the '=' sign, and save the file.
4. Restart the Oracle Identity Analytics application in a new browser window to view your changes.

An excerpt from the rbacxmessages.properties file follows:

```
####################################################
#    Identity Certification
####################################################

idc.error.errorUsersRequired = No Users found in this certification...
idc.error.errorRolesRequired = No Roles found in this certification...
idc.error.updateCommentsFailed = Unable to update the comments!
idc.error.noEndPointsFound = No EndPoints found in this certification!
idc.error.selectCertification = Please select at least one certification
first!
idc.error.selectReport = Please select a report first!
idc.error.reportError = This report has no pages!
idc.error.selectDate = Please select date values!
idc.error.selectMonths = Please select months values!
idc.error.selectYear = Please select year values!
idc.error.selectSeconds = Please select seconds values!
idc.error.selectMinutes = Please select minute values!
idc.error.selectHours = Please select hour values!
idc.error.errorAlphanumericCharactersRequired = Certification name must contain alphanumeric  /
characters!
idc.error.deleteFailed = Unable to delete certification job!
idc.error.unableToAdd = Unalbe to add certification job!
idc.error.checkHighlightedFields = Please check the highlighted fields!
idc.error.selectBusinessUnit = Please select a business unit!
```

Configuring the Maximum Number of Identity Certification Records That Should Display in the UI

You can configure the maximum number of records that users see onscreen when viewing certifications on the My Certifications screen by defining batch sizes. Batch sizes for data owner certifications, user entitlement certifications, and resource entitlement certifications can be configured, but role entitlement certifications cannot.

If UI batch sizes are set too high, long load times can result. If set too low, end-users will need to request pages more often, which can also result in delays.

Note - To improve identity certification performance, server-side batching can be configured. See *Configuring Identity Certification Settings on the Server* for information.

To Modify Identity Certification Batch Sizes in the UI

1. Open the `idc.properties` file located in `$RBACX_WAR/conf` to configure batch sizes for user entitlement certifications and resource entitlement certifications.

   (To configure batch sizes for data owner certifications, open the `idc-context.xml` file located in `$RBACX_WAR/conf`.)
2. Scroll to the section that says `IDC UI batch sizes`.
3. Find the correct configuration key, then change the value.

   See the following examples:

   - For user entitlement certifications, find the `com.vaau.rbacx.idc.ui.usersBatchSize` key and change the numeric value up or down.

- For resource entitlement certifications, find the `com.vaau.rbacx.idc.ui.resourcesBatchSize` key and change the numeric value up or down.

4. Save the file.
5. Restart the Oracle Identity Analytics application in a new browser window to view your changes.

APPENDIX A

Oracle Waveset Sample Workflows

Oracle Waveset Sample Workflows Introduction

This appendix includes sample Oracle Waveset workflows that can be used to facilitate the integration of Oracle Waveset (previously Sun Identity Manager) with Oracle Identity Analytics (previously Sun Role Manager).

Note - Do not use the sample workflows included with Oracle Waveset 8.1.1. Instead, either use the updated workflows available in Waveset 8.1.1-Patch 1, or the updated workflows available in this appendix.

For more information about these workflows, see the “Configuring the Maximum Number of Identity Certification Records That Should Display in the UI” on page 78 chapter.

```
<?xml version=’1.0’ encoding=’UTF-8’?>
<!DOCTYPE Waveset PUBLIC ’waveset.dtd’ ’waveset.dtd’>
<Waveset>

<!--
   Workflow Processes related to integration with Role Manager.
-->

<!--============================================================================
  Check SRM Integration
    Invokes WorkflowServices to determine if SRM integration has been
    configured.  Returns the boolean value in the isSRMIntegrated variable.
=============================================================================-->
<Configuration name=’Check SRM Integration’>
  <Extension>
    <WFProcess name=’Check SRM Integration’>
      <Variable name=’isSRMIntegrated’ output=’true’/>

      <Activity name=’start’>
        <Transition to=’Test SRM Integration’/>
      </Activity>

      <Activity name=’Test SRM Integration’>
```

```
        <Action application=’com.waveset.provision.WorkflowServices’>
          <Argument name=’op’ value=’isSRMIntegrated’/>
          <Return from=’isSRMIntegrated’ to=’isSRMIntegrated’/>
        </Action>

        <Transition to=’end’/>
      </Activity>

      <Activity name=’end’/>
    </WFProcess>
  </Extension>
</Configuration>

<!--============================================================================
  Merge SRM Role Assignments
    If SRM is integrated and the UserView option ’getRuleDrivenRoleManagerRoles’
    is set to ’true’, this process will retrieve the list of roles to be
    automatically assigned by SRM configured rules.  This list of roles will be
    merged with the IdM assigned roles into the UserView.
=============================================================================-->
<Configuration name=’Merge SRM Role Assignments’>
  <Extension>
    <WFProcess name=’Merge SRM Role Assignments’>
      <Variable name=’user’ input=’true’/>

      <Variable name=’refreshedUser’ output=’true’>
        <Comments>
          The refreshed view of the user after merging SRM rule-based role
          assignments.
        </Comments>
      </Variable>

      <Variable name=’isSRMIntegrated’>
        <Comments>
          This is a boolean indicating if SRM is integrated.
        </Comments>
      </Variable>

      <Variable name=’previewedRoles’>
        <Comments>
          The list of roles SRM is reporting should be assigned to the user.
        </Comments>
      </Variable>

      <Variable name=’wsResult’>
        <Comments>
          This is a GenericObject holding the result of calls to SRM web services.
        </Comments>
      </Variable>

      <Activity name=’start’>
        <Transition to=’Check SRM Integration’/>
      </Activity>

      <Activity name=’Check SRM Integration’>
        <Action process=’Check SRM Integration’>
          <Return from=’isSRMIntegrated’ to=’isSRMIntegrated’/>
        </Action>
        <Transition to='Check SRM Role Assignments'>
          <and>
            <isTrue><ref>isSRMIntegrated</ref></isTrue>
            <isTrue>
              <get>
                <ref>user</ref>
                <s>getRuleDrivenRoleManagerRoles</s>
              </get>
            </isTrue>
          </and>
        </Transition>
        <Transition to='Set Refreshed User'/>
      </Activity>

      <Activity name='Set Refreshed User'>
        <Action name='Set Refreshed User'>
          <set name='refreshedUser'>
            <ref>user</ref>
          </set>
        </Action>

        <Transition to='end'/>
      </Activity>

      <Activity name='Check SRM Role Assignments'>
        <Variable name='inputParams'>
          <map>
            <s>user</s>
            <ref>user</ref>
          </map>
        </Variable>

        <Action name='Get SRM Role Assignments'
                application='com.waveset.provision.WorkflowServices'>
          <Argument name='op' value='callWebService'/>
          <Argument name='wsAction' value='Preview'/>
          <Argument name='wsObjType' value='Role'/>
          <Argument name='resource' value='Sun Role Manager Web Services'/>
          <Argument name='parameters' value='$(inputParams)'/>
          <Argument name='catch' value='cepException'/>
          <Return from='wsResult' to='wsResult'/>
        </Action>

        <Action name='Set previewed roles'>
          <set name='previewedRoles'>
            <get>
              <ref>wsResult</ref>
              <s>PreviewRoles</s>
            </get>
          </set>
        </Action>

        <Transition to='Merge SRM Role Assignments'/>
      </Activity>

      <Activity name='Merge SRM Role Assignments'>
        <Variable name='view'>
          <Comments>
            The refreshed view of the user after merging SRM rule-based role
```

```
            assignments.
          </Comments>
        </Variable>

        <Action name='Set Role Manager Roles'>
          <invoke name='put'>
            <ref>user</ref>
            <s>waveset.roleManagerRoles</s>
            <ref>previewedRoles</ref>
          </invoke>
        </Action>

        <Action name='Set SelectAll'>
          <Comments>
            Indicate that any resources resulting from preview role assignments
            should be provisioned.
          </Comments>

          <invoke name='put'>
            <ref>user</ref>
            <s>update.selectAll</s>
            <s>true</s>
          </invoke>
        </Action>

        <Action name='Update Role Assignments'
                application='com.waveset.session.WorkflowServices'>
          <Argument name='op' value='refreshView'/>
          <Argument name='view' value='$(user)'/>
          <Return from='view' to='refreshedUser'/>
        </Action>

        <Transition to='end'/>
      </Activity>

      <Activity name='end'/>
    </WFProcess>
  </Extension>
</Configuration>

<!--============================================================================
  Create SRM User
    If SRM is integrated, this process will invoke the a create SRM user action
    based on UserView attributes.
==============================================================================-->
<Configuration name='Create SRM User'>
  <Extension>
    <WFProcess name='Create SRM User'>
      <Variable name='user' input='true'>
        <Comments>
          The GenericObject containing the UserView for the account to be created.
        </Comments>
      </Variable>

      <Variable name='isSRMIntegrated'>
        <Comments>
          This is a boolean indicating if SRM is integrated.
        </Comments>
      </Variable>
      <Activity name='start'>
        <Transition to='Check SRM Integration'/>
      </Activity>

      <Activity name='Check SRM Integration'>
        <Action process='Check SRM Integration'>
          <Return from='isSRMIntegrated' to='isSRMIntegrated'/>
        </Action>

        <Transition to='Create SRM User'>
          <isTrue>
            <ref>isSRMIntegrated</ref>
          </isTrue>
        </Transition>
        <Transition to='end'/>
      </Activity>

      <Activity name='Create SRM User'>
        <Variable name='inputParams'>
          <map>
            <s>user</s>
            <ref>user</ref>
          </map>
        </Variable>

        <Action application='com.waveset.provision.WorkflowServices'>
          <Argument name='op' value='callWebService'/>
          <Argument name='wsAction' value='Create'/>
          <Argument name='wsObjType' value='User'/>
          <Argument name='resource' value='Sun Role Manager Web Services'/>
          <Argument name='parameters' value='$(inputParams)'/>
          <Argument name='catch' value='cepException'/>
        </Action>

        <Transition to='end'/>
      </Activity>

      <Activity name='end'/>
    </WFProcess>
  </Extension>
</Configuration>

<!--============================================================================
  Update SRM User
    If SRM is integrated, this process will invoke the an update SRM user action
    based on UserView attributes.
=============================================================================-->
<Configuration name='Update SRM User'>
  <Extension>
    <WFProcess name='Update SRM User'>
      <Variable name='user' input='true'>
        <Comments>
          The GenericObject containing the UserView for the account to be updated.
        </Comments>
      </Variable>

      <Variable name='updateRequested'>
        <Comments>
          Variable indicating whether an update is requested.  This variable
          will default to ’true’.  OIA/SRM can set this to ’false’ to avoid
          an additional update to that system when a role membership request
          originates in OIA/SRM.
        </Comments>
      </Variable>

      <Variable name=’isSRMIntegrated’>
        <Comments>
          A boolean indicating if SRM is integrated.
        </Comments>
      </Variable>

      <Activity name=’start’>
        <Transition to=’Check Update Requested’/>
      </Activity>

      <Activity name=’Check Update Requested’>
        <Action name=’Set Update Requested’>
          <set name=’updateRequested’>
            <cond>
              <ref>user.viewOptions.updateOIA</ref>
              <isTrue>
                <ref>user.viewOptions.updateOIA</ref>
              </isTrue>
              <i>1</i>
            </cond>
          </set>
        </Action>

        <Transition to=’Check SRM Integration’>
          <isTrue>
            <ref>updateRequested</ref>
          </isTrue>
        </Transition>

        <Transition to=’end’/>
      </Activity>

      <Activity name=’Check SRM Integration’>
        <Action process=’Check SRM Integration’>
          <Return from=’isSRMIntegrated’ to=’isSRMIntegrated’/>
        </Action>

        <Transition to=’Update SRM User’>
          <isTrue>
            <ref>isSRMIntegrated</ref>
          </isTrue>
        </Transition>
        <Transition to=’end’/>
      </Activity>

      <Activity name=’Update SRM User’>
        <Variable name=’inputParams’>
          <map>
            <s>user</s>
            <ref>user</ref>
          </map>
        </Variable>
```

```
        <Action application=’com.waveset.provision.WorkflowServices’>
          <Argument name=’op’ value=’callWebService’/>
          <Argument name=’wsAction’ value=’Update’/>
          <Argument name=’wsObjType’ value=’User’/>
          <Argument name=’resource’ value=’Sun Role Manager Web Services’/>
          <Argument name=’parameters’ value=’$(inputParams)’/>
          <Argument name=’catch’ value=’cepException’/>
        </Action>

        <Transition to=’end’/>
      </Activity>

      <Activity name=’end’/>
    </WFProcess>
  </Extension>
</Configuration>


<!--============================================================================
  Rename SRM User
    If SRM is integrated, a rename SRM user action.
=============================================================================-->
<Configuration name=’Rename SRM User’>
  <Extension>
    <WFProcess name=’Rename SRM User’>
      <Variable name=’accountId’ input=’true’>
        <Comments>
          The name of the user to rename.
        </Comments>
      </Variable>

      <Variable name=’newAccountId’ input=’true’>
        <Comments>
          The new user name.
        </Comments>
      </Variable>

      <Variable name=’isSRMIntegrated’>
        <Comments>
          This is a boolean indicating if SRM is integrated.
        </Comments>
      </Variable>

      <Activity name=’start’>
        <Transition to=’Check SRM Integration’/>
      </Activity>

      <Activity name=’Check SRM Integration’>
        <Action process=’Check SRM Integration’>
          <Return from=’isSRMIntegrated’ to=’isSRMIntegrated’/>
        </Action>

        <Transition to=’Rename SRM User’>
          <isTrue>
            <ref>isSRMIntegrated</ref>
          </isTrue>
        </Transition>
        <Transition to=’end’/>
      </Activity>

      <Activity name='Rename SRM User'>
        <Variable name='inputParams'>
          <map>
            <s>accountId</s>
            <ref>accountId</ref>
            <s>newAccountId</s>
            <ref>newAccountId</ref>
          </map>
        </Variable>

        <Action application='com.waveset.provision.WorkflowServices'>
          <Argument name='op' value='callWebService'/>
          <Argument name='wsAction' value='Rename'/>
          <Argument name='wsObjType' value='User'/>
          <Argument name='resource' value='Sun Role Manager Web Services'/>
          <Argument name='parameters' value='$(inputParams)'/>
          <Argument name='catch' value='cepException'/>
        </Action>

        <Transition to='end'/>
      </Activity>

      <Activity name='end'/>
    </WFProcess>
  </Extension>
</Configuration>

<!--================================================================================
  Delete SRM User
    If SRM is integrated, a delete SRM user action.
==============================================================================-->
<Configuration name='Delete SRM User'>
  <Extension>
    <WFProcess name='Delete SRM User'>
      <Variable name='accountId' input='true'>
        <Comments>
          The ID of the user to delete.
        </Comments>
      </Variable>

      <Variable name='options' input='true'>
        <Comments>
          Variables indicating whether or not the IdM user is to be deleted.
          The SRM user will be deleted if the IdM user will be.
        </Comments>
      </Variable>

      <Variable name='isSRMIntegrated'>
        <Comments>
          A boolean indicating if SRM is integrated.
        </Comments>
      </Variable>

      <Activity name='start'>
        <Transition to='Check SRM Integration'>
          <!--
            The conditions here are different from the ones used for disable/enable
            but they are exactly the conditions used by the DeProvision subtask,
            reused here for consistency
          -->
          <or>
            <contains>
              <ref>options.targets</ref>
              <s>Lighthouse</s>
            </contains>
            <isTrue>
              <ref>options.deleteUser</ref>
            </isTrue>
            <isTrue>
              <ref>options.forceDelete</ref>
            </isTrue>
          </or>
        </Transition>
        <Transition to=’end’/>
      </Activity>

      <Activity name=’Check SRM Integration’>
        <Action process=’Check SRM Integration’>
          <Return from=’isSRMIntegrated’ to=’isSRMIntegrated’/>
        </Action>

        <Transition to=’Delete SRM User’>
          <isTrue>
            <ref>isSRMIntegrated</ref>
          </isTrue>
        </Transition>
        <Transition to=’end’/>
      </Activity>

      <Activity name=’Delete SRM User’>
        <Variable name=’inputParams’>
          <map>
            <s>accountId</s>
            <ref>accountId</ref>
          </map>
        </Variable>

        <Action application=’com.waveset.provision.WorkflowServices’>
          <Argument name=’op’ value=’callWebService’/>
          <Argument name=’wsAction’ value=’Delete’/>
          <Argument name=’wsObjType’ value=’User’/>
          <Argument name=’resource’ value=’Sun Role Manager Web Services’/>
          <Argument name=’parameters’ value=’$(inputParams)’/>
          <Argument name=’catch’ value=’cepException’/>
        </Action>

        <Transition to=’end’/>
      </Activity>

      <Activity name=’end’/>
    </WFProcess>
  </Extension>
</Configuration>

<!--============================================================================
  Disable SRM User
```

```
    If SRM is integrated, a disable SRM user action.
================================================================================-->
<Configuration name='Disable SRM User'>
  <Extension>
    <WFProcess name='Disable SRM User'>
      <Variable name='accountId' input='true'>
        <Comments>
          The ID of the user to disable.
        </Comments>
      </Variable>

      <Variable name='options' input='true'>
        <Comments>
          Variables indicating whether or not the IdM user is to be disabled.
          The SRM user will be disabled if the IdM user will be.
        </Comments>
      </Variable>

      <Variable name='isSRMIntegrated'>
        <Comments>
          A boolean indicating if SRM is integrated.
        </Comments>
      </Variable>

      <Activity name='start'>
        <Transition to='Check SRM Integration'>
          <or>
            <isnull>
              <ref>options</ref>
            </isnull>
            <!--
              The viewers no longer put simple lists of strings into options.targets,
              however, customizations might still do.
            -->
            <contains>
              <ref>options.targets</ref>
              <s>Lighthouse</s>
            </contains>
            <contains>
              <dolist name='resInfo'>
                <ref>options.targets</ref>
                <invoke name='getResourceName'>
                  <ref>resInfo</ref>
                </invoke>
              </dolist>
              <s>Lighthouse</s>
            </contains>
          </or>
        </Transition>
        <Transition to='end'/>
      </Activity>

      <Activity name='Check SRM Integration'>
        <Action process='Check SRM Integration'>
          <Return from='isSRMIntegrated' to='isSRMIntegrated'/>
        </Action>

        <Transition to='Disable SRM User'>
          <isTrue>
```

```
            <ref>isSRMIntegrated</ref>
          </isTrue>
        </Transition>
        <Transition to=’end’/>
      </Activity>

      <Activity name=’Disable SRM User’>
        <Variable name=’inputParams’>
          <map>
            <s>accountId</s>
            <ref>accountId</ref>
          </map>
        </Variable>

        <Action application=’com.waveset.provision.WorkflowServices’>
          <Argument name=’op’ value=’callWebService’/>
          <Argument name=’wsAction’ value=’Disable’/>
          <Argument name=’wsObjType’ value=’User’/>
          <Argument name=’resource’ value=’Sun Role Manager Web Services’/>
          <Argument name=’parameters’ value=’$(inputParams)’/>
          <Argument name=’catch’ value=’cepException’/>
        </Action>

        <Transition to=’end’/>
      </Activity>

      <Activity name=’end’/>
    </WFProcess>
  </Extension>
</Configuration>

<!--============================================================================
  Enable SRM User
    If SRM is integrated, a enable SRM user action.
=============================================================================-->
<Configuration name=’Enable SRM User’>
  <Extension>
    <WFProcess name=’Enable SRM User’>
      <Variable name=’accountId’ input=’true’>
        <Comments>
          The ID of the user to enable.
        </Comments>
      </Variable>

      <Variable name=’options’ input=’true’>
        <Comments>
          Variables indicating whether or not the IdM user is to be enabled.
          The SRM user will be enabled if the IdM user will be.
        </Comments>
      </Variable>

      <Variable name=’isSRMIntegrated’>
        <Comments>
          A boolean indicating if SRM is integrated.
        </Comments>
      </Variable>

      <Activity name=’start’>
        <Transition to=’Check SRM Integration’>
          <or>
            <isnull>
              <ref>options</ref>
            </isnull>
            <!--
              The viewers no longer put simple lists of strings into options.targets,
              however, customizations might still do.
            -->
            <contains>
              <ref>options.targets</ref>
              <s>Lighthouse</s>
            </contains>
            <contains>
              <dolist name='resInfo'>
                <ref>options.targets</ref>
                <invoke name='getResourceName'>
                  <ref>resInfo</ref>
                </invoke>
              </dolist>
              <s>Lighthouse</s>
            </contains>
          </or>
        </Transition>
        <Transition to='end'/>
      </Activity>

      <Activity name='Check SRM Integration'>
        <Action process='Check SRM Integration'>
          <Return from='isSRMIntegrated' to='isSRMIntegrated'/>
        </Action>

        <Transition to='Enable SRM User'>
          <isTrue>
            <ref>isSRMIntegrated</ref>
          </isTrue>
        </Transition>
        <Transition to='end'/>
      </Activity>

      <Activity name='Enable SRM User'>
        <Variable name='inputParams'>
          <map>
            <s>accountId</s>
            <ref>accountId</ref>
          </map>
        </Variable>

        <Action application='com.waveset.provision.WorkflowServices'>
          <Argument name='op' value='callWebService'/>
          <Argument name='wsAction' value='Enable'/>
          <Argument name='wsObjType' value='User'/>
          <Argument name='resource' value='Sun Role Manager Web Services'/>
          <Argument name='parameters' value='$(inputParams)'/>
          <Argument name='catch' value='cepException'/>
        </Action>

        <Transition to='end'/>
      </Activity>
```

```
      <Activity name='end'/>
    </WFProcess>
  </Extension>
</Configuration>

<!--===============================================================================
  Notify Reconcile Response
===============================================================================-->

<!-- note that this does not use wstype='ProvisioningTask' we leave it
  == as TaskDefinition so that the ProcessViewer can find it, it is expected
  == to do its own authorization
-->
<TaskDefinition name='Create SRM User Reconcile Response Workflow'
                authType='ReconAdminTask'
                taskType='Workflow'
                executor='com.waveset.workflow.WorkflowExecutor'
                syncControlAllowed='true'
                execMode='sync '
                visibility='invisible'>
  <Extension>
    <WFProcess name='Create SRM User Reconcile Response Workflow'>
      <Comments>
        A sample per-account action.
        Creates an SRM user when responsePerformed is CREATE_NEW_USER.
      </Comments>

      <Variable name='accountId' input='true'>
        <Comments>
          The account to which a response was just applied.
        </Comments>
      </Variable>

      <Variable name='resourceId' input='true'>
        <Comments>
          The object id of the resource  where the account resides.
        </Comments>
      </Variable>

      <Variable name='resourceName' input='true'>
        <Comments>
          The name of the resource resource  where the account resides.
        </Comments>
      </Variable>

      <Variable name='userId' input='true'>
        <Comments>
          The object id of the Lighthouse user associated with the account.
          If no user is associated with the account, this is null.
        </Comments>
      </Variable>

      <Variable name='userName'>
        <Comments>
          The name of the Lighthouse user associated with the account.
          If no user is associated with the account, this is null.
        </Comments>
      </Variable>
```

```
<Variable name='initialSituation' input='true'>
  <Comments>
    The situation that was initially discovered for the account,
    triggering the response.
    The value is a valid message key.
  </Comments>
</Variable>

<Variable name='responseSuccess'>
  <Comments>
    A boolean indicating whether the response was successful.
  </Comments>
</Variable>

<Variable name='finalSituation'>
  <Comments>
    The situation of the account after the response was performed.
    The value is a valid message key.
    If the account no longer exists - on the resource and in Lighthouse -
    the value is null.
  </Comments>
</Variable>

<Variable name='responsePerformed'>
  <Comments>
    The id of the response performed.  One of the following:
      LINK_ACCOUNT
      UNLINK_ACCOUNT
      CREATE_ACCOUNT
      DELETE_ACCOUNT
      DISABLE_ACCOUNT
      CREATE_NEW_USER
      DO_NOTHING
  </Comments>
</Variable>

<Variable name='responsePerformedText'>
  <Comments>The string representation of the response performed</Comments>
</Variable>

<Variable name='isSRMIntegrated'>
  <Comments>
    This is a boolean indicating if SRM is integrated.
  </Comments>
</Variable>

<Variable name='user'>
  <Comments>
    The GenericObject containing the UserView for the account to be created.
  </Comments>
</Variable>

<Activity name='start'>
  <Transition to='end'>
    <Comments>
      Do nothing if no response was attempted or the action is not
      CREATE_NEW_USER.
    </Comments>
    <or>
      <isFalse>
        <ref>responseSuccess</ref>
      </isFalse>
      <or>
        <isnull>
          <ref>responsePerformed</ref>
        </isnull>
        <neq>
          <ref>responsePerformed</ref>
          <s>CREATE_NEW_USER</s>
        </neq>
      </or>
    </or>
  </Transition>
  <Transition to=’Check SRM Integration’/>
</Activity>

<Activity name=’Check SRM Integration’>
  <Action process=’Check SRM Integration’>
    <Return from=’isSRMIntegrated’ to=’isSRMIntegrated’/>
  </Action>

  <Transition to=’Get Userview’>
    <isTrue>
      <ref>isSRMIntegrated</ref>
    </isTrue>
  </Transition>
  <Transition to=’end’/>
</Activity>

<Activity name=’Get Userview’>
  <Variable name=’view’>
    <Comments>
      Local variable to hold view for view that gets checked out
    </Comments>
  </Variable>

  <Action application=’com.waveset.session.WorkflowServices’>
    <Argument name=’op’ value=’getView’/>
    <Argument name=’type’ value=’User’/>
    <Argument name=’NoFetch’ value=’true’/>
    <Argument name=’NoViolationForm’ value=’true’/>
    <Argument name=’id’>
      <ref>userName</ref>
    </Argument>
    <Return from=’view’ to=’user’/>
  </Action>

  <Transition to=’Create SRM User’/>
</Activity>

<Activity name=’Create SRM User’>
  <Variable name=’inputParams’>
    <map>
      <s>user</s>
      <ref>user</ref>
    </map>
  </Variable>
```

```
        <Action application=’com.waveset.provision.WorkflowServices’>
          <Argument name=’op’ value=’callWebService’/>
          <Argument name=’wsAction’ value=’Create’/>
          <Argument name=’wsObjType’ value=’User’/>
          <Argument name=’resource’ value=’Sun Role Manager Web Services’/>
          <Argument name=’parameters’ value=’$(inputParams)’/>
          <Argument name=’catch’ value=’cepException’/>
        </Action>

        <Transition to=’end’/>
      </Activity>

      <Activity name=’end’/>
    </WFProcess>
  </Extension>
  <MemberObjectGroups>
    <ObjectRef type=’ObjectGroup’ id=’#ID#Top’ name=’Top’/>
  </MemberObjectGroups>
</TaskDefinition>

</Waveset>
```