

# Developing OIM Plug-ins

## Overview

This example is used to demonstrate how to develop OIM Plug-ins using JDeveloper.

## Objective

The objective of this example is to develop the following OIM plug-ins using JDeveloper:

- An Oracle Identity Manager request status changed plug-in. This sample plug-in displays the message “Request <request ID> failed.” on the console when a request’s status changes to “Request Failed”.
- An Oracle Identity Manager kernel event handler. This sample event handler displays the message “Finalizing a create user orchestration process with process ID: <orchestration process ID>” on the console during the finalization stage of a create user operation.
- An Oracle Identity Manager scheduled task. This sample scheduled task displays the message “Running the OIM Sample Scheduled Task...” on the console when run as a scheduled job.

## Scenario

Angel works as a system engineer for Mydo Main Corporation. She is responsible for integrating identity and access management systems within the company. One such task is extending the functionalities of the Oracle Identity Manager via developing Oracle Identity Manager plug-ins. In order to do that, she decides to write simple Oracle Identity Manager plug-ins to get familiar with how to develop an Oracle Identity Manager plug-in.

## Prerequisites

Before starting this tutorial, the following tasks need to be completed:

- Oracle Identity Manager 11g should be installed.
- The files provided as part of this example should be downloaded to the OIM server machine. Following section explains the files provided in the example.
- Environment variable OIM\_ORACLE\_HOME (i.e.: .../.../middleware/Oracle\_IDM1) should be set to the home directory of the Oracle Identity Manager.
- Environment variable WL\_HOME (i.e.: .../.../middleware/wlserver\_10.3) should be set to the home directory of the Weblogic server.
- wfullclient.jar should be generated.
- Apache Ant 1.7 or later should be installed and environment variable ANT\_HOME should be set.
- JDK 1.6 or later should be installed and environment variable JAVA\_HOME should be set.

From [here](#) download the zip file that contains the files that are required for this example.

## Files in the example

/OIMPluginSamples

- OIMPluginSamples.jws – The JDeveloper workspace.
- /Plugin
  - o build.xml – The Ant build file.
  - o EventHandlers.xml – The EventHandlers.xml that defines the example kernel event handler.
  - o OIMSampleScheduledTask.xml – The XML file that defines the example scheduled task.
  - o Plugin.jpr – The JDeveloper project.
  - o plugin.xml – The plugin.xml that defines all the new plug-ins.
- /src
  - o oracle.iam.sample.eventhandler.MyUserCreateFinalizationHandler – The source code of the example kernel event handler.
  - o oracle.iam.sample.plugin.RequestFailedChangeEvent – The source code for the example plug-in.
  - o oracle.iam.sample.scheduledtask.SampleScheduledTask – The source code for the example scheduled task.

## Steps to build and deploy the plug-ins

To build and deploy all three plug-ins (The request status changed plug-in, the kernel event handler, and the scheduled task):

1. Modify the following segments in the build.xml file to provide the Oracle Identity Manger username that will be used to register the plug-ins (OIM.Username), the weblogic server name on which Oracle Identity Manager is running (WL.ServerName), the weblogic server username (WL.Username), and the server URL (ServerURL):

```
<property name="OIM.Username" value="xelsysadm"/>
```

```
<property name="ServerURL" value="t3://localhost:7001"/>
```

```
<property name="WL.ServerName" value="AdminServer"/>
```

```
<property name="WL.Username" value="weblogic"/>
```

2. Use the command “ant register” to build and deploy the plug-ins to Oracle Identity Manager. It will register the plug-ins and upload the necessary metadata files needed for the plug-ins.
3. Restart the OIM managed server for the plug-ins to take effect.

If you want to remove the plug-ins from Oracle Identity Manager, you can use the command “ant unregister”. It will unregister the plug-ins and remove all the metadata files that were uploaded by the “ant register” command.