HP Select Identity Software

Connector for Oracle® E-Business Suite 11i

Connector Version: 2.01

Installation and Configuration Guide

Document Release Date: September 2007
Software Release Date: September 2007

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

This product includes software developed by the Apache Software Foundation (http://www.apache.org/). Portions Copyright © 1999-2003 The Apache Software Foundation. All rights reserved.

Select Identity uses software from the Apache Jakarta Project including:

- Commons-beanutils
- Commons-collections
- Commons-logging
- Commons-digester
- Commons-httpclient
- Element Construction Set (ecs)
- Jakarta-poi
- Jakarta-regexp
- Logging Services (log4j)

Additional third party software used by Select Identity includes:

- JasperReports developed by SourceForge
- iText (for JasperReports) developed by SourceForge
- BeanShell
- Xalan from the Apache XML Project
- Xerces from the Apache XML Project
- Java API for XML Processing from the Apache XML Project
- SOAP developed by the Apache Software Foundation
- JavaMail from SUN Reference Implementation
- Java Secure Socket Extension (JSSE) from SUN Reference Implementation
- Java Cryptography Extension (JCE) from SUN Reference Implementation
- JavaBeans Activation Framework (JAF) from SUN Reference Implementation

- OpenSPML Toolkit from OpenSPML.org
- JGraph developed by JGraph
- Hibernate from Hibernate.org
- BouncyCastle engine for keystore management, bouncycastle.org

Trademark Notices

Support

You can visit the HP software support web site at:

http://www.hp.com/go/hpsoftwaresupport

HP Software online support provides an efficient way to access interactive technical support tools. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require an active support contract.

To find more information about support access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Contents

1 Documentation Map

This chapter describes the organization of HP Select Identity connector documentation and provides necessary information on how to use the documentation set to install and configure the connectors.

Figure 1 illustrates the documentation map for Select Identity connector. For a list of available product documentation, refer to the Table 1.

Figure 1 Documentation Map

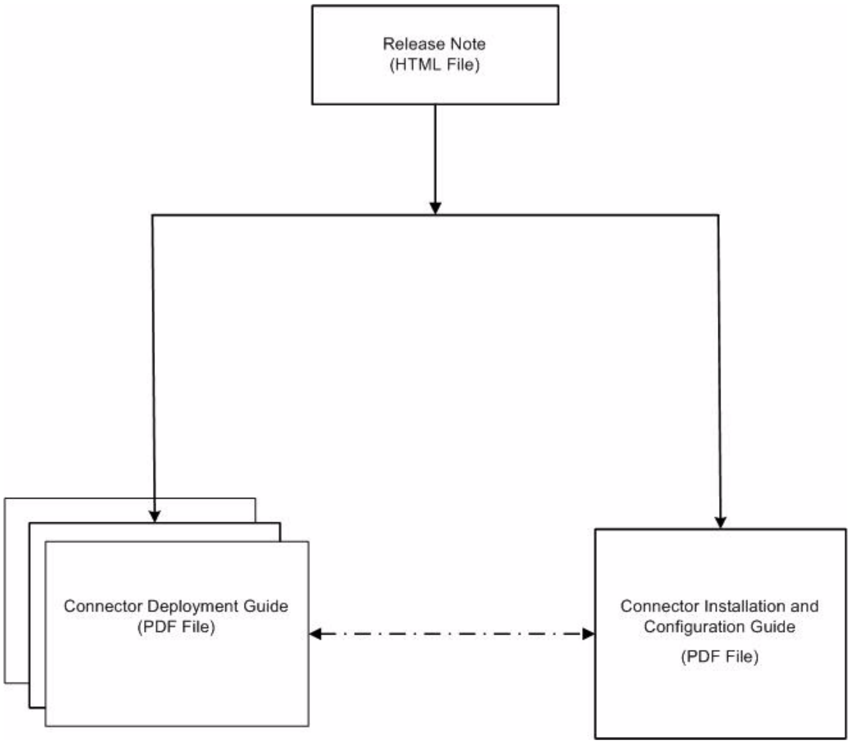


Table 1 Connector Documentation

Document Title and Filename	Contents	Location
Release Note `Oracle 11i Connector v2.0 Release Note.htm`	This file contains necessary information on new features of the connector, enhancements, known problems or limitations, and support information.	`/Docs/` subdirectory under the connector directory.
Connector Deployment Guide (for Select Identity 4.20) `connector_deploy_SI4.20.pdf`	Connector deployment guides provide detailed information on: • Deploying a connector on an application server. • Configuring a connector with Select Identity. Refer to these guides when you need generic information on connector installation.	`/Docs/` root directory on the product's CD media.
Connector Deployment Guide (for Select Identity 4.10-4.13) `connector_deploy_SI4.13.pdf`		
Connector Deployment Guide (for Select Identity 4.0-4.01) `connector_deploy_SI4.pdf`		
Connector Deployment Guide (for Select Identity 3.3.1) `connector_deploy_SI3.3.1.pdf`		
Connector Installation and Configuration Guide `Oracle 11i_guide.pdf`	Connector installation and configuration guide provides installation instructions for a specific connector. It contains resource specific configuration details.	`/Docs/` subdirectory under the connector directory.

2 Introduction

This chapter gives an overview of the HP Select Identity connector for Oracle E-Business Suite 11i. An HP Select Identity connector for Oracle E-Business Suite 11i enables you to provision users and manage identities on Oracle E-Business Suite 11i. At the end of this chapter, you will be able to know about:

- The benefits of HP Select Identity.
- The role of a connector.
- The connector for Oracle E-Business Suite 11i.

About HP Select Identity

HP Select Identity provides a new approach to identity management. Select Identity helps you automate the process of provisioning and managing user accounts and access privileges across platforms, applications, and corporate boundaries. Select Identity communicates with the enterprise information system through connectors, and automates the tasks of identity management. The enterprise information system, which is also referred to as **resource**, can be a database, a directory service, or an ERP package, among many others.

About Connectors

You can establish a connection between a resource and Select Identity by using a connector. A connector is resource specific. The combination of Select Identity and connector helps you perform a set of tasks on the resource to manage identity. A connector can be **unidirectional** or **bidirectional**. A unidirectional connector helps you manage identities from Select Identity, but if any change takes place in resource, it cannot communicate that back to Select Identity. On the other hand, a bidirectional connector can reflect the changes made on resource back to Select Identity. This property of bidirectional connectors is known as **reverse synchronization**.

About Oracle 11i Connector

The Oracle E-Business Suite 11i connector — hereafter referred to as the Oracle 11i connector — enables Select Identity to provision users on Oracle E-Business Suite 11i systems. The Oracle 11i connector enables Select Identity to perform the following provisioning tasks on Oracle E-Business Suite 11i for Oracle 11i Application users (foundation users):

- Add, update, and remove users

- Retrieve user attributes
- Enable and disable users
- Verify a user’s existence
- Change user passwords
- Reset user passwords
- Expire passwords
- Retrieve all entitlements
- Retrieve a list of supported user attributes
- Grant and revoke entitlements to and from users

The following operations are supported in reverse provisioning for Oracle 11i Application users (foundation users):

- Add, update, and remove users
- Assign and revoke entitlements to and from users

In reverse provisioning, user password reset is not supported. In place of real password, a default password is sent over reconciliation process (which can be configured in agent configuration file), and later it can be reset from Select Identity by using reset password operation.

The following functions are supported for employee reverse provisioning from Oracle 11i HRMS:

- Creation of a new employee
- Changes in employee attributes
- Termination of an employee
- Reverse Termination of employee (re-employing a terminated employee)

The Oracle 11i connector is a two-way connector and provides an agent that can send changes made to data in Oracle Application and Human Resources systems to Select Identity.

The connector does not perform any forward operation Oracle Human Resources system. Instead, information is sent to Select Identity by the agent when an employee is created, modified, terminated, or reverse terminated in Oracle Human Resources.

The Oracle 11i connector can be used with Select Identity 3.3.1-4.20.

Overview of Installation Tasks

Before you start installing the connector, you must ensure that system requirements and all the installation prerequisites are met. Refer to the Table 2 for an overview of installation tasks.

Table 2 Organization of Tasks

Task Number	Task Name	Reference
1	Install the connector on the Select Identity server.	See Installing the Connector on page 13.
	— Meet the system requirements.	See System Requirements on page 15.
	— Extract the contents of the Schema file (file that contains the mapping files for the connector) to location on the Select Identity server.	See Extracting Contents of the Schema File on page 15.
	— Install the Resource Adapter Archive (RAR) file of the connector.	See Installing the Connector RAR on page 15.
2	Install the agent on the resource system.	See Installing the Agent on page 17.
3	Configure the connector with Select Identity.	See Configuring the Connector with Select Identity on page 71.

3 Installing the Connector

This chapter elaborates the procedure to install Oracle 11i connector on Select Identity server. At the end of this chapter, you will know about

- Software requirements to install the Oracle 11i connector.
- Prerequisite conditions to install Oracle 11i connector.
- Procedure to install Oracle 11i connector.

Oracle 11i Connector Files

The Oracle 11i connector is packaged in the following files in the Oracle 11i directory on the Select Identity Connector CD:

Table 3 Oracle 11i Connector Files

Serial Number	File Name	Description
1	• oraerp_420.rar for WebSphere • oraerp_420WL9.rar for WebLogic	The Resource Adapter Archive (RAR) file contains the connector binaries.
2	schema.jar	The Schema file contains the mapping files that contain attribute information of Oracle E-Business Suite 11i. It contains the following files: • ORAERP-11-5-9.xml and ORAERP-11-5-10.xml — map the Select Identity fields to the Oracle fields, which enables Select Identity to provision data in Oracle E-Business Suite systems. Use one of these files that matches the installed version of Oracle Application. • ORAERPEMP-11-5-9.xml and ORAERPEMP-11-5-10.xml — map the Select Identity attributes to the Oracle Human Resources employee attributes. Use one of these mapping files if you are provisioning to Oracle Human Resources systems (this file also enables you to map attributes on Oracle E-Business Suite 11i servers). Use the file that matches the installed version of Oracle E-Business Suite 11i. • oracle11ierp.xsl — maps attributes on the Oracle 11i Human Resources server and Oracle Applications server to attributes on the Select Identity server. This file is used by the agent during reverse synchronization.
3	oraerpsecattr.zip	It contains the SQL Script for security attributes.
4	oraerpagent.zip	The zip file contains the executable to install the agent.

System Requirements

The Oracle 11i connector is supported in the following environment:

Table 4 Platform Matrix for the Oracle 11i Connector

Select Identity Version	Application Server	Database
3.3.1	WebLogic 8.1.4 on Windows 2003	SQL Server 2000
	WebSphere 5.1.1 on HP-UX 11i	Oracle 9i
	WebSphere 5.1.1 on Windows 2003	Oracle 9i
4.0-4.20	The Oracle 11i connector is supported on all the platform configurations of Select Identity 4.0-4.20.	

This connector supports Oracle E-Business Suite 11.5.9 on HP-UX 11i and 11.5.10 on Windows 2000, HP-UX 11i, and Solaris 9. For secure communication, this connector supports secure JDBC. See Configuring a Secure JDBC Connection Pool on page 79 for configuration information.

Extracting Contents of the Schema File

The Schema file of the connector contains necessary mapping information to map resource attributes to Select Identity. Extract contents of the `schema.jar` file to a directory that is in the application server `CLASSPATH`. Refer to the *HP Select Identity Connector Deployment Guide* for detailed instruction to extract contents of the Schema file.

Installing the Connector RAR

To install the RAR file of the connector (such as `oraerp_420.rar`) on the Select Identity server, you must copy the file to a local subdirectory on the Select Identity server, and then deploy on the application server. Refer to the *HP Select Identity Connector Deployment Guide* for detailed information on deploying a RAR file on an application server.

While deploying the RAR on WebSphere, enter the JNDI Pool Name as **`eis/ORAERP`**.

4 Installing the Agent

This chapter gives an overview of the agent for Oracle 11i connector and the procedure to install the agent on an Oracle E-Business Suite 11i system. At the end of the chapter, you will be able to know about:

- The role of an agent.
- The procedure to install the agent.

About the Agent

The Oracle 11i connector communicates to the Oracle E-Business Suite 11i resource with the help of the agent. For forward (Select Identity to Oracle 11i) operations, the connector communicates with the agent and agent performs the provisioning on the resource. Agent sends back any changes made on Oracle 11i to Select Identity web service in the form of SPML requests. The agent is packaged with the file `oraerpagent.zip`. The agent uses Oracle Application Alert system to monitor events on Oracle E-Business Suite 11i and sends the change back to Select Identity.

Installing the Agent on Oracle E-Business Suite 11i Server

After you install the Oracle 11i connector on the Select Identity server, you must deploy SQL scripts, which comprise the agent, on the target Oracle system. The following sections describe the steps you must take to deploy the SQL scripts.

Perform the following procedures to install the agent:

1. Install the SQL Scripts
2. Create Oracle Alerts
3. Create Alerts for Employee Reverse Synchronization

Install the SQL Scripts

Perform the following steps to install the scripts on a target Oracle system:

1. If required, set the environment variable for the database. An environment setting script is located in the Oracle Application home directory. You can get this location for $APPL_TOP or %APPL_TOP% from your Oracle Application administrator. (The environment may not be set correctly if you not logged on as an Oracle Application user.)
2. Create a subdirectory on the resource server where you can place the Oracle SQL scripts.

3 Extract the contents of the `oraerpsecattr.zip` and `oraerpagent.zip` files to the subdirectory created.

4 Run the `sioerp.sql` script on SQL *Plus as a system user, as shown below:

```
Sqlplus system/manager
@ sioraerp.sql;
```

This command creates a custom schema called `sioraerp` that contains the following tables used in reverse provisioning.

- `SIORAERP.USER_Request` — This table contains individual events recorded by Oracle alert actions for a user. Oracle alert stores user IDs for application users and person IDs for employees, an employee or user indicator, and the requested action type. The action type is A for add action, M for modify action, or T for terminate. The `SIORAERP.USER_Request` table is shown below:

```
REQUEST_NUM NUMBER NOT NULL,
ID NUMBER NOT NULL, /* User ID for app user and
* Person ID for employee */
USER_TYPE CHAR NOT NULL, /* E for Employee and U for User */
ACTION_TYPE CHAR NOT NULL /* A for Add, M for Modify,
* T for terminate */
```

- `SIORAERP.PROVISION_REQUEST` — This table contains the Select Identity provisioning request for a user. At regular intervals, the requests in the `USER_REQUESTS` table are combined to create reverse provisioning requests for a user in the `PROVISION_REQUESTS` table. The `SIORAERP.PROVISION_REQUEST` table is shown below:

```
REQUEST_NUM NUMBER NOT NULL,
SI_REQUEST_ID NUMBER, /* Not currently used */
CREATION_DATE DATE,
LAST_SENT_DATE DATE, /* Date and time SPML message
                      * sent last time*/
USER_NAME VARCHAR2(64),
RESOURCE_NAME VARCHAR2(64),
USER_ID NUMBER NOT NULL, /* User ID for app user and
                          * PersonID for employee */
USER_TYPE CHAR NOT NULL, /* E for Employee and
                          * U for User */
ADD_ACTION NUMBER, /* 1 or 0 */
MODIFY_ACTION NUMBER, /* 1 or 0 */
TERMINATE_ACTION NUMBER, /* 1 or 0 */
REQ_STATUS NUMBER, /* 0= ready for process, 1=in process,
                    * 2=completed, -1 = error */
SPML_MESSAGE VARCHAR2(4000), /* SPML message going out */
RESPONSE VARCHAR2(4000), /* Response from SI */
ERROR_MSG VARCHAR2(4000), /* Error Message */
RETRY NUMBER /* Retry count for this message */
```

- `SIORAERP.OWMINFO` — This table contains Oracle Wallet information. Oracle Wallet contains the security certificates used for secure communication. The Wallet information contains the path to the Wallet directory and password. If the Wallet information is valid and the URL is specified as HTTPS, a secure (SSL) channel is used to communicate with the Select Identity Web Service.

```
OWMPATH VARCHAR(256) NOT NULL, /* Wallet Directory */
OWMPASS VARCHAR(64) /* Wallet password */
```

5 Run the `oraerp_secAttr.sql` script in SQL *Plus as an application user. This script creates PL/SQL packages, which handle securing attribute updates from the Select Identity Oracle 11i connector. An example of the command to run the script on SQL *Plus is given below:

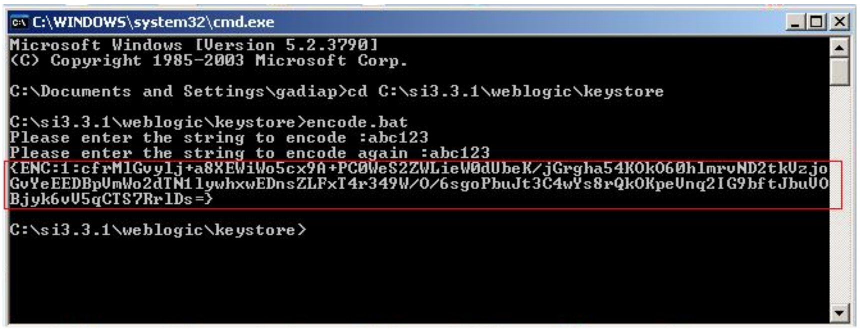
```
Sqlplus apps/apps
@ oraerp_secAttr.sql;
```

6 Run the `oraerp_spml.sql` script in SQL *Plus as an application user. This script creates `PL/SQL Provision_Service` packages, which handle SPML message handling for the agent.

7 Edit the `SIConfig.sql` script to enter values for the Select Identity server, administrator's user name, password, and the resource ID for the application user reverse synchronization and employee reverse synchronization. You must modify the `SIHOST`, `SIADMIN`, `SIPWD`, `SIFNDRESOURCE`, and `SIEMPRESOURCE` parameters.

> If you wish to generate an SPML file that can be used to import a large number of users from Oracle 11i into Select Identity, you can edit the `SIConfig.sql` script as described in Generating Files for the Auto Discovery Function on page 80.

To encrypt the password, run `encode.bat` (on Windows) or `encode.sh` (on UNIX), which is provided in the `weblogic/keystore` subdirectory in the Select Identity home directory. This utility prompts you for the password to encrypt and will generate the encrypted password. You must copy the entire encrypted password, as shown here:



C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\gadiap>cd C:\si3.3.1\weblogic\keystore

C:\si3.3.1\weblogic\keystore>encode.bat
Please enter the string to encode :abc123
Please enter the string to encode again :abc123
<ENC:1:cfrMlGvylj+a8XEWiWo5cx9A+PC0WeS2ZWLieW0dUbeK/jGrgha54KOkO60hlmrvND2tkVzjo
GvYeEEDBpVmWo2dTN1lywhxwEDnsZLFxT4r349W/O/6sgoPbuJt3C4wYs8rQkOKpeVnq2IG9bftJbuVO
Bjyk6vV5qCTS7RrlDs=>

C:\si3.3.1\weblogic\keystore>
```

The application user reverse synchronization resource is created in the `ORAERP-11-5-x.xml` mapping file. The employee reverse synchronization resource is created in the `ORAERPEMP-11-5-x.xml` mapping file.

8 *Wallet configuration for SSL communication only:*
After Oracle Wallet configuration is complete, run the `updateowminfo.sql` script as the sioraerp or apps user. Be sure to run the script from the Select Identity installation directory because the script loads the `SIConfig.sql` file to get the URL of the Select Identity server.

`updateowminfo.sql` prompt for the Oracle Wallet Manager (OWM) directory and the password. It then stores the information in the `SIORAERP.OWMINFO` table. When the wallet path is entered, the path information is stored in the table as file:$PATH, such as file:c:\owm or file:/app/owm. The wallet path is the directory located in the Oracle database server, not the application forms server.

9 Copy the SQL scripts from the `foundation` and `employee` subdirectories to the Select Identity directory created in step 2 (such as `C:\app\selectid` on Windows).

On the Select Identity server, modify the `ORAERP-11-5-x.xml`, `ORAERPEMP-11-5-x.xml`, and `oracle11ierp.xsl` files to verify that the parameters specified in the `SIConfig.sql` script are the same. If new attributes were added to the `SIConfig.sql` script, these attributes must also be added to the XSL and XML files.

Oracle Application Agent Provision_service Package

To enable reverse synchronization on Oracle Application or Oracle Human Resources systems, you must understand the Provision_service PL/SQL package. This package handles the SPML message processing that occurs when the agent sends requests to the Select Identity server.

The Provision_service package provides the following data types:

- provision_service.atttab_type — attribute name/value table
- provision_service.val_list_type — entitlement list

The package provides the following procedures:

- invoke_add_request — adds a user
- invoke_modify_request — modifies a user
- invoke_delete_request — deletes a user
- invoke_enable_svc_request — assigns a user to a Select Identity Service
- invoke_disable_svc_request — removes a user from a Select Identity Service
- invoke_reset_password_request — resets a password for a user
- invoke_enable_user_request — enables a user

Oracle Application Agent Provision_service Package

To enable reverse synchronization on Oracle Application or Oracle Human Resources systems, you must understand the Provision_service PL/SQL package. This package handles the SPML message processing that occurs when the agent sends requests to the Select Identity server.

The Provision_service package provides the following data types:

- provision_service.atttab_type — attribute name/value table
- provision_service.val_list_type — entitlement list

The package provides the following procedures:

- invoke_add_request — adds a user
- invoke_modify_request — modifies a user
- invoke_delete_request — deletes a user
- invoke_enable_svc_request — assigns a user to a Select Identity Service
- invoke_disable_svc_request — removes a user from a Select Identity Service
- invoke_reset_password_request — resets a password for a user
- invoke_enable_user_request — enables a user

- invoke_disable_user_request — disables a user
- invoke_terminate_user_request — terminates a user
- invoke_update_password_request — updates a user's password

The following list gives a description of all the arguments used in these procedures:

url — URL of the Select Identity server.

request_id — Request ID.

admin_id — ID of a Select Identity user under which the service request will be performed.

admin_pwd — The password of the Select Identity user.

service_name — The name of the Select Identity Service for which this request is performed.

resource_id — The name of the Select Identity resource for which this request is performed.

attrs — List of attributes.

entitlements — List of entitlements to be added during user creation.

addentitlements — List of entitlements to be added during user modification.

deleteentitlements — List of entitlements to be removed during user modification.

Here is an example for adding a new user:

```
DECLARE
    attab provision_service.atttab_type;
    add_entlist provision_service.val_list_type;
BEGIN
    attab('UserName') := 'CHPARK69';
    attab('Password') := 'abc123';
    attab('Company') := 'MJM';
    attab('Email') := 'chpark@mjm.com';
    attab('Owner') := 'CUST';
    attab('Person') := '';
    attab('Desc') := 'Description';
    attab('Customer') := '';
    attab('Supplier') := '';
    attab('Fax') := '8225267883';
    attab('Days') := '20';
    attab('Accesses') := '20';
    attab('StartDate') := '2005-2-1';
    attab('EndDate') := '2005-3-1';
add_entlist := provision_service.val_list_type();
provision_service.add_entitlement(add_entlist, 'CM_NORWAY');
provision_service.invoke_add_request('http://helix:7003/lmz/webservice', '12345',
'sisa', 'abc123', '', 'vis', attab, add_entlist);
END;
/
```

Event Alert Definitions

The agent uses the Oracle Applications Alert system to implement event monitoring on the Oracle system. When information about an Oracle user or employee changes, Oracle Alert provides immediate action based on the criteria defined by the user.

With Oracle Alert, the Oracle system can be monitored without using external programs, and it can be customized easily to suit the needs of the organization for user and employee event monitoring. For the reverse synchronization agent, event alerts are used. An event alert immediately notifies you of activity in the database as it occurs. When you create an event alert, you specify the following:

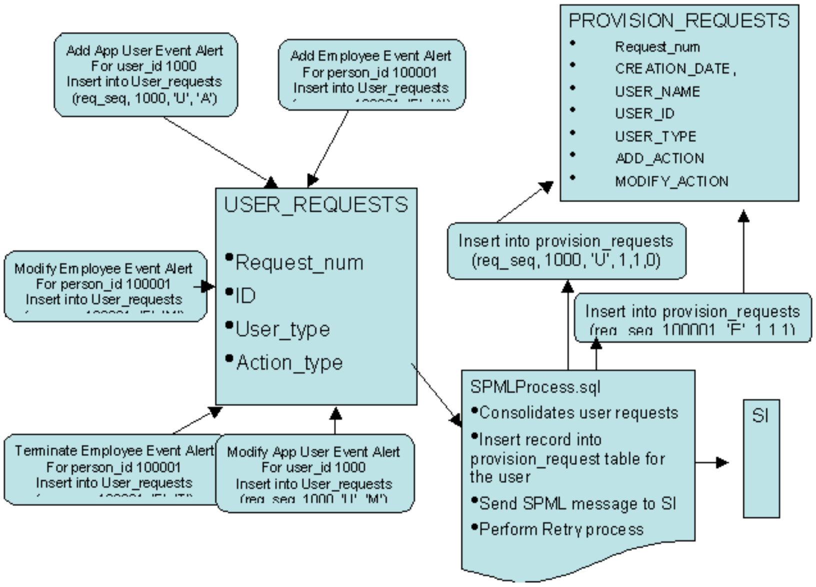
- A database event that you want to monitor (an insert or update to a specific database table).
- A SQL Select statement that retrieves specific database information as a result of the database event.

- Actions that you want Oracle Alert to perform as a result of the database event. An action can entail sending someone an email message, running a concurrent program, running an operating script, or running a SQL statement script. You specify the actions that the Oracle Alert should perform in an action set.

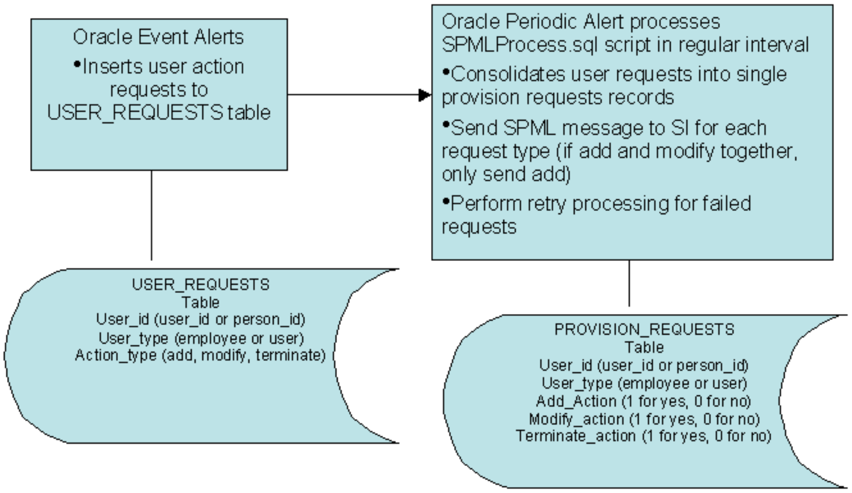
To create an event alert, you must perform the following tasks:

1. Define the database events that will trigger the alert.
2. Specify the details of the alert.
3. Define actions for the alert.
4. Create action sets containing the actions you want the alert to perform.

The Oracle 11i connector agent uses Oracle Event alerts to capture individual event actions on a user (add, modify, terminate) and stores that individual event in a temporary processing table (sioraerp.user_requests table). At regular intervals, a periodic alert processes a consolidation processing script (`SPMLProcess.sql`) that combines all actions for a user and creates corresponding the SPML message to be sent to Select Identity.


Add App User Event Alert
For user_id 1000
Insert into User_requests
(req_seq, 1000, 'U', 'A')
Add Employee Event Alert
For person_id 100001
Insert into User_requests
PROVISION_REQUESTS
• Request_num
• CREATION_DATE ,
• USER_NAME
• USER_ID
• USER_TYPE
• ADD_ACTION
• MODIFY_ACTION
USER_REQUESTS
•Request_num
•ID
•User_type
•Action_type
Modify Employee Event Alert
For person_id 100001
Insert into User_requests
Insert into provision_requests
(req_seq, 1000, 'U', 1,1,0)
Insert into provision_requests
(req_seq, 100001, 'E', 1,1,1)
SPMLProcess.sql
•Consolidates user requests
•Insert record into provision_request table for the user
•Send SPML message to SI
•Perform Retry process
SI
Terminate Employee Event Alert
For person_id 100001
Insert into User_requests
Modify App User Event Alert
For user_id 1000
Insert into User_requests
(req_seq, 1000, 'U', 'M')

The following diagram illustrates the data flow to and from the agent:


Oracle Event Alerts
•Inserts user action requests to USER_REQUESTS table
Oracle Periodic Alert processes SPMLProcess.sql script in regular interval
•Consolidates user requests into single provision requests records
•Send SPML message to SI for each request type (if add and modify together, only send add)
•Perform retry processing for failed requests
USER_REQUESTS Table
User_id (user_id or person_id)
User_type (employee or user)
Action_type (add, modify, terminate)
PROVISION_REQUESTS Table
User_id (user_id or person_id)
User_type (employee or user)
Add_Action (1 for yes, 0 for no)
Modify_action (1 for yes, 0 for no)
Terminate_action (1 for yes, 0 for no)

The following sections describe how to create event alerts based on whether an application user or employee is provisioned.

Create Oracle Alerts

An Oracle Application user uses the Oracle Application system to perform required business processing. An Oracle Application user differs from an Oracle Human Resources employee in the that the Application user actively uses Oracle Application to perform business tasks. An Oracle Human Resources employee is simply stored in the Oracle Human Resources system to represent an employee in the organization. However, an Oracle Human Resources employee can be an Oracle Application user, and vice versa.

The agent monitors the following events. If any of these events occur, the agent send the new data to Select Identity (reverse synchronization). The user name is the key field that identifies each Application user. The user name must be unique in the Oracle Application system, and the user name is not case sensitive (the agent sends the user name is all uppercase letters).

- New Application user
- Change in user attributes
- Change in user responsibilities
- Change in securing attributes for a user
- Termination of a user

The following user attributes are synchronized with Select Identity:

- Description — A description of the user.
- Employee ID — ID of the user if the user ia also an employee in the Oracle Human Resources system.
- Customer ID — ID of the customer contact defined in Oracle Application
- Supplier ID — ID of the supplier contact defined in Oracle Application
- E-Mail — Email address for the user
- Fax — Fax number for the user

- Password Access Days — Maximum number of days between password changes. A pop-up window prompts an Application user to change her or his password after the maximum number of days you specify has elapsed.
- Password Accesses — Maximum allowed number of sign-ons to Oracle Applications between password changes. A pop-up window prompts an application user to change her or his password after the maximum number of accesses you specify has elapsed. Password Access Days and Password Accesses are mutually exclusive. Only one of these attributes may be used for the Oracle Application user.
- Start Date — Start date for the user. The user cannot sign onto Oracle Application before the start date and after the end date. The default for the start date is the current date.
- End Date — End date for the user. The user cannot sign onto Oracle Application after the end date. If the end date is not specified, the user name is valid indefinitely. An Application user cannot be deleted from Oracle Application because this information provides an audit trail. An Oracle Application user can be deactivated at any time by setting the End Date to the current date. To reactivate a user, change the End Date to a date after the current date, or clear the End Date field.

The user events are monitored by Oracle event alerts. Alerts are defined to monitor user creation, user modification, responsibility modification, and securing attributes changes. Additional events can be monitored by custom alert definitions. The alert action scripts call the main processing SQL scripts

with parameters retrieved from the alert action. The processing script performs further data processing on the attributes and calls the Provision_service PL/SQL package to send SPML message to Select Identity.

Scripts are provided by the agent for the following events:

- **Add Application User**
  Application Name: Application Object Library
  Table Name: FND_USER
  Event to Monitor: After Insert
  Select SQL Script: AddUserSelect.sql
  Action SQL Script: AddUserAction.sql
- **Modify Application User**
  Application Name: Application Object Library
  Table Name: FND_USER
  Event to Monitor: After Update
  Select SQL Script: ModifyUserSelect.sql
  Action SQL Script: ModifyUserAction.sql
- **Update Responsibility**
  Application Name: Application Object Library
  Table Name: FND_USER_RESP_GROUPS
  Event to Monitor: After Insert After Update
  Select SQL Script: ModifyRespSelect.sql
  Action SQL Script: ModifyRespAction.sql
- **Modify Securing Attributes**
  Application Name: Oracle Common Modules -AK
  Table Name: AK_WEB_USER_SEC_ATTR_VALUES
  Event to Monitor: After Insert After Update
  Select SQL Script: ModifySecAttrSelect.sql
  Action SQL Script: ModifySecAttrAction.sql

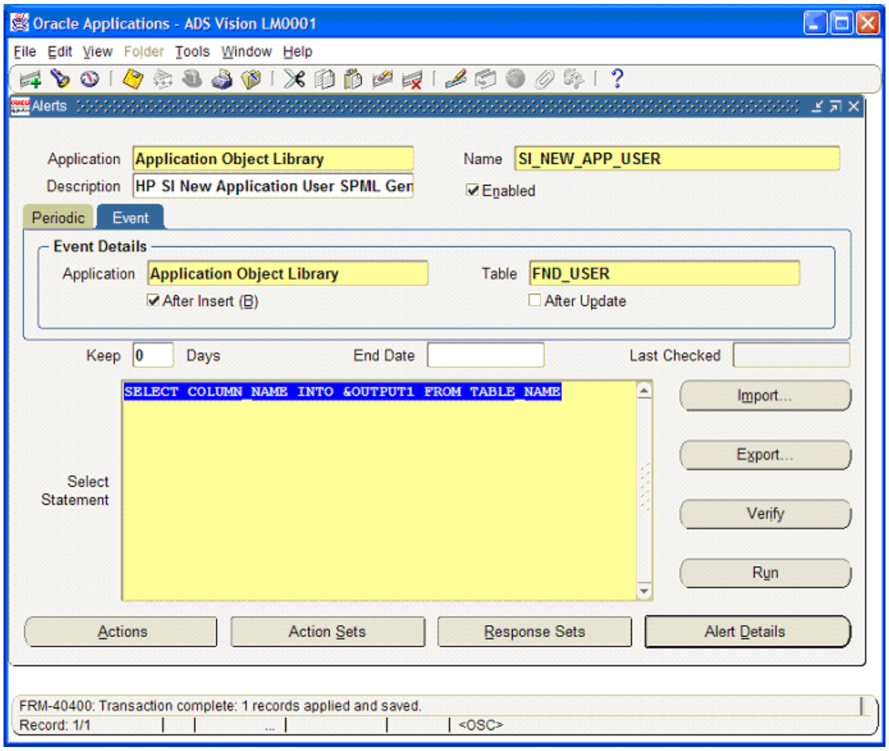
The following sections describe how to create an alert in Oracle Application for each of these events.

> Instead of importing the alert scripts from file, it is possible to copy and paste to update the alert actions and select statements. On some platforms (such as Windows), importing causes in fewer issues due to embedded special characters.

Define a New Application User Alert

Perform the following steps to define an alert for a new Application user:

1 Display the alert definition window from the Oracle Application Alert Manager.



Application: Application Object Library

Name: SI_NEW_APP_USER

Description: HP SI New Application User SPML Gen

☑ Enabled

Event Details

Application: Application Object Library

Table: FND_USER

☑ After Insert (B)

□ After Update

Keep: 0 Days

End Date:

Last Checked:

Select Statement:

```
SELECT COLUMN_NAME INTO &OUTPUT1 FROM TABLE_NAME
```

```
FRM-40400: Transaction complete: 1 records applied and saved.
Record: 1/1 <OSC>
```

2 Click the **Event** tab to define the event alert.

3 Enter `Application Object Library` in the Application field.

4 Enter `FND_USER` in the Table field.

5 Select the **After Insert** option. Make sure that the **After Update** option is deselected.

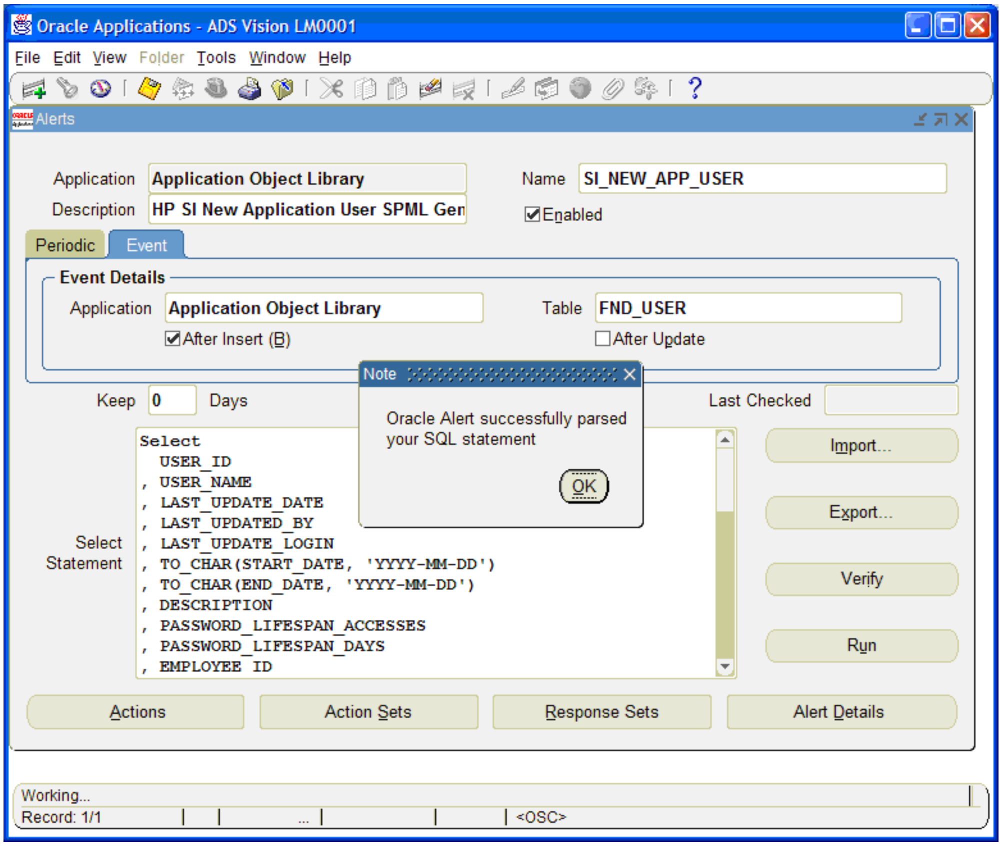
6 Enter the alert name and description.

7 Save the alert.

8 Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `AddUserSelect.sql` script.

9 Click **OK** on the file upload window.

10 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.



Application: Application Object Library

Description: HP SI New Application User SPML Gen

Name: SI_NEW_APP_USER

☑ Enabled

Event Details

Application: Application Object Library

Table: FND_USER

☑ After Insert (B)

□ After Update

Keep: 0 Days

Last Checked:

Note: Oracle Alert successfully parsed your SQL statement

Select Statement:

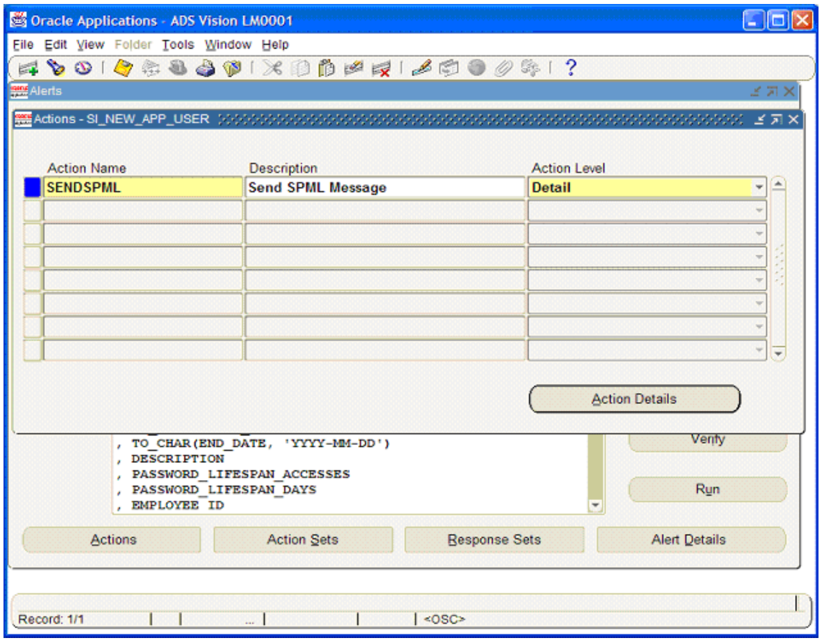
```
Select
  USER_ID
, USER_NAME
, LAST_UPDATE_DATE
, LAST_UPDATED_BY
, LAST_UPDATE_LOGIN
, TO_CHAR(START_DATE, 'YYYY-MM-DD')
, TO_CHAR(END_DATE, 'YYYY-MM-DD')
, DESCRIPTION
, PASSWORD_LIFESPAN_ACCESSES
, PASSWORD_LIFESPAN_DAYS
, EMPLOYEE_ID
```

Working...

Record: 1/1 | | ... | | | <OSC>

11 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

12 Enter the action name, such as **SENDSPML**. Select **Detail** as the action level, then save the alert.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_NEW_APP_USER
Action Details
Verify
Run
Actions
Action Sets
Response Sets
Alert Details

Action Name	Description	Action Level
SENDSPML	Send SPML Message	Detail

```
, TO_CHAR(END_DATE, 'YYYY-MM-DD')
, DESCRIPTION
, PASSWORD_LIFESPAN_ACCESSES
, PASSWORD_LIFESPAN_DAYS
, EMPLOYEE_ID
```

Record: 1/1 | | ... | | | <OSC>

13 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

14 Select **SQL Statement Script** for the Action Type.

15 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

16 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `AddUserAction.sql` script. This populates the window with the content of the script.

17 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

18 Save the alert.

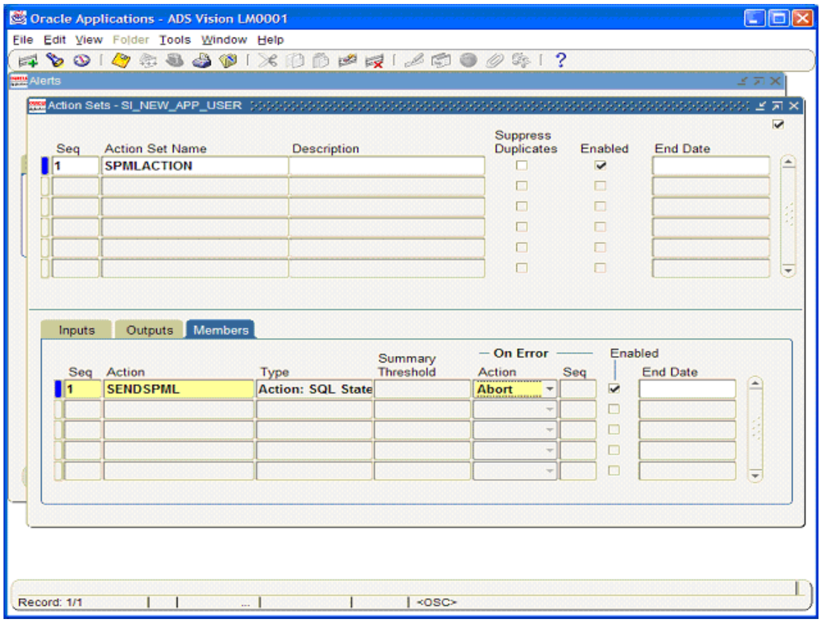
19 Close the Alert Details definition window and Action window and go to main alert definition window.

20 Click the **Action Sets** button.

21 Enter the action set name, such as **`SPMLACTION`**, and save.

22 Click the **Members** tab.

23 Choose the action defined above and save the alert.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
?
Alerts
Action Sets - SI_NEW_APP_USER
Inputs
Outputs
Members
Record: 1/1
...
<OSC>

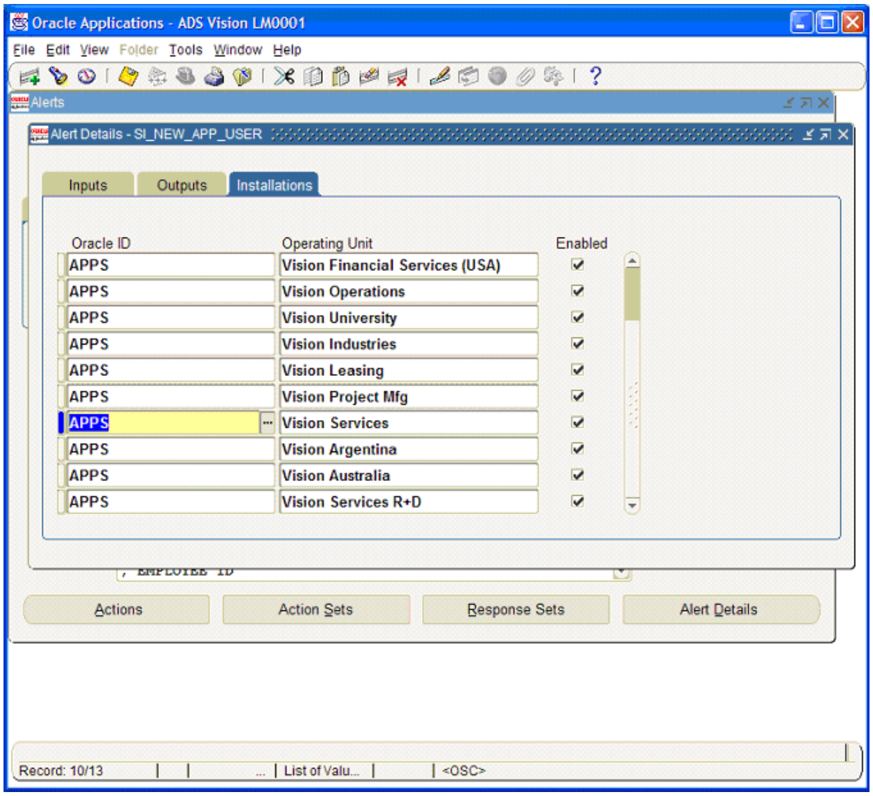
Seq	Action Set Name	Description	Suppress Duplicates	Enabled	End Date
1	SPMLACTION		☐	☑	

				On Error			
Seq	Action	Type	Summary Threshold	Action	Seq	Enabled	End Date
1	SENDSPML	Action: SQL State		Abort		☑	

24 Close the Action Sets window and click **Alert Details** to display the Alert Details window.

25 In the alert details window, click the **Installations** tab.

26 Enter the application user ID (**apps**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Alert Details - SI_NEW_APP_USER
Inputs
Outputs
Installations
Actions
Action Sets
Response Sets
Alert Details
Record: 10/13
List of Valu...
<OSC>

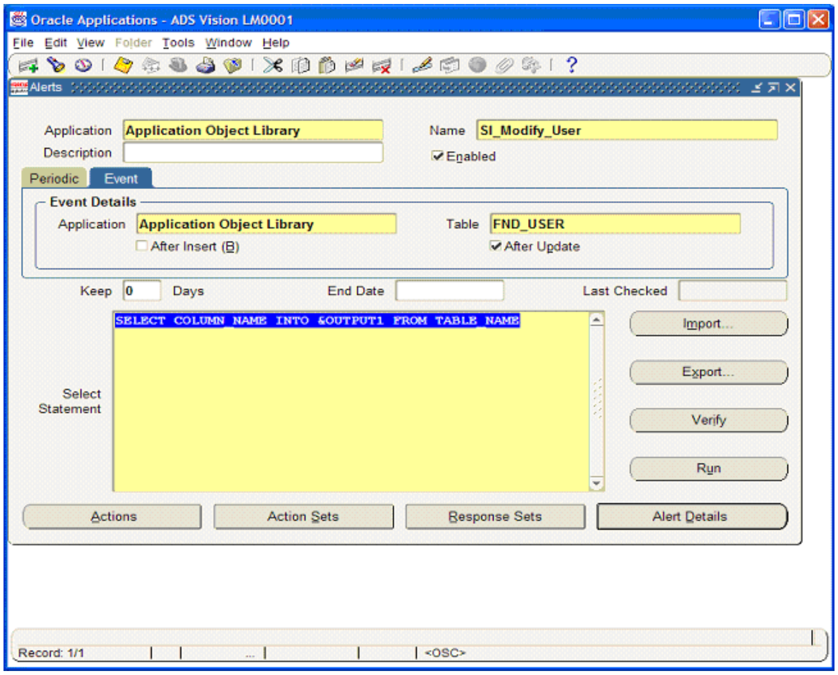
Oracle ID	Operating Unit	Enabled
APPS	Vision Financial Services (USA)	☑
APPS	Vision Operations	☑
APPS	Vision University	☑
APPS	Vision Industries	☑
APPS	Vision Leasing	☑
APPS	Vision Project Mfg	☑
APPS	Vision Services	☑
APPS	Vision Argentina	☑
APPS	Vision Australia	☑
APPS	Vision Services R+D	☑

27 Save the alert. The alert is now enabled and active.

Define an Alert for Application User Modifications

Perform the following steps to define an alert that is triggered when an Application user is modified:

1 Display the alert definition window from the Oracle Application Alert Manager.



Application: Application Object Library

Name: SI_Modify_User

Description:

☑ Enabled

Event Details

Application: Application Object Library

Table: FND_USER

□ After Insert (B)

☑ After Update

Keep: 0 Days

End Date:

Last Checked:

Select Statement:

```
SELECT COLUMN_NAME INTO &OUTPUT1 FROM TABLE_NAME
```

2 Click the **Event** tab to define the event alert.

3 Enter **`Application Object Library`** in the Application field.

4 Enter **`FND_USER`** in the Table field.

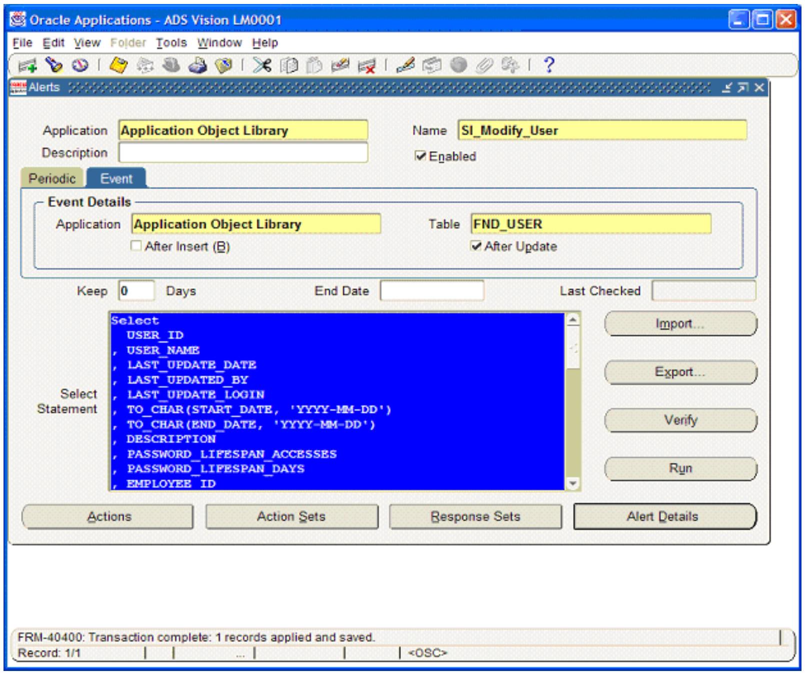
5 Select the **After Update** option. Make sure that the **After Insert** option is deselected.

6 Enter the alert name and description.

7 Save the alert.

8 Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyUserSelect.sql` script.

9 Click **OK** on the file upload window.



Application: Application Object Library

Name: SI_Modify_User

Description:

☑ Enabled

Event Details

Application: Application Object Library

Table: FND_USER

☐ After Insert (B)

☑ After Update

Keep: 0 Days

End Date:

Last Checked:

Select Statement:

```
Select
  USER_ID
, USER_NAME
, LAST_UPDATE_DATE
, LAST_UPDATED_BY
, LAST_UPDATE_LOGIN
, TO_CHAR(START_DATE, 'YYYY-MM-DD')
, TO_CHAR(END_DATE, 'YYYY-MM-DD')
, DESCRIPTION
, PASSWORD_LIFESPAN_ACCESSES
, PASSWORD_LIFESPAN_DAYS
, EMPLOYEE_ID
```

FRM-40400: Transaction complete: 1 records applied and saved.

Record: 1/1 | | ... | | | <OSC>

10 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.

11 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

12 Enter the action name, such as **`SENDSPML`**. Select **Detail** as the action level, then save the alert.

13 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

14 Select **SQL Statement Script** for the Action Type.

15 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

16 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyUserAction.sql` script. This populates the window with the content of the script.

17 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

18 Save the alert.

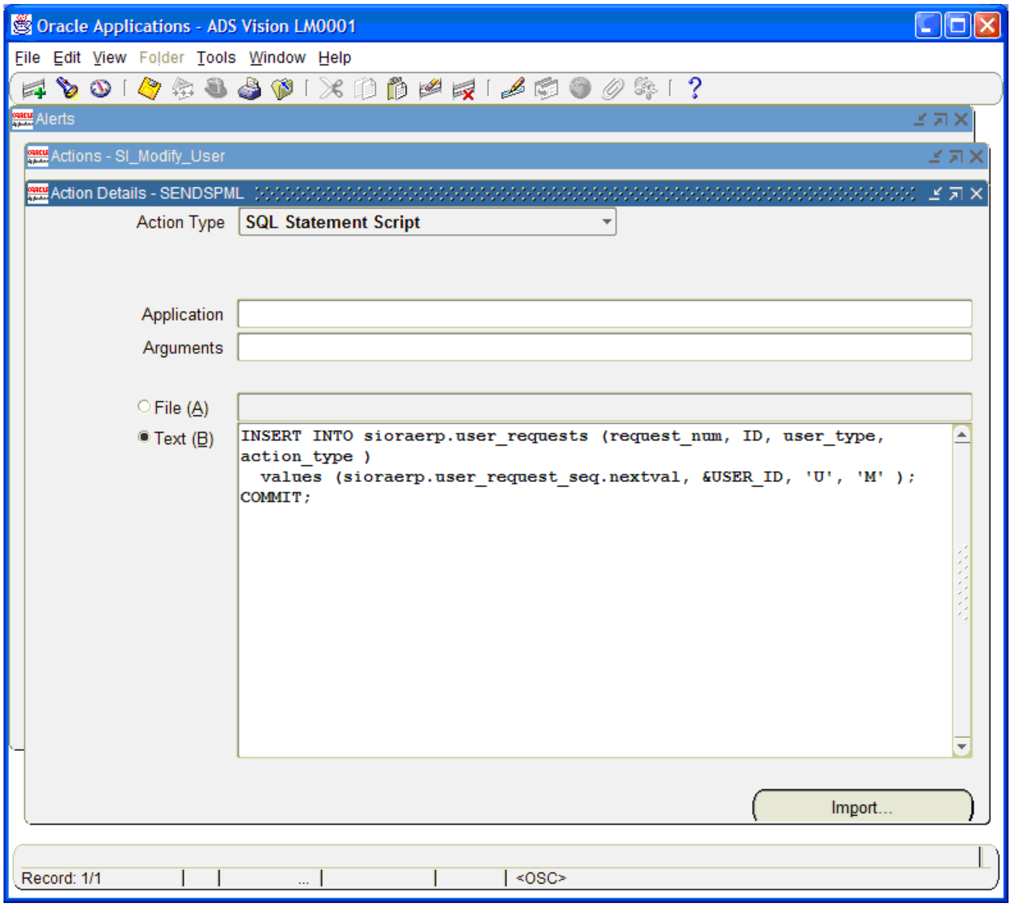
19 Close the Alert Details definition window and Action window and go to main alert definition window.

20 Click the **Action Sets** button.

21 Enter the action set name, such as **`SPMLACTION`**, and save.

22 Click the **Members** tab.

23 Choose the action defined above and save the alert.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_Modify_User
Action Details - SENDSPML
Import...
Record: 1/1
<OSC>

Action Type: SQL Statement Script

Application:

Arguments:

○ File (A)

● Text (B)

```
INSERT INTO sioraerp.user_requests (request_num, ID, user_type,
action_type )
  values (sioraerp.user_request_seq.nextval, &USER_ID, 'U', 'M' );
COMMIT;
```

24 Close the Action Sets window and click **Alert Details** to display the Alert Details window.

25 In the alert details window, click the **Installations** tab.

26 Enter the application user ID (**`apps`**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).

27 Save the alert. The alert is now enabled and active.

Define an Alert for Application User Responsibility Modifications (v11.5.9)

Perform the following steps on Oracle 11.5.9 to define an alert that is triggered when an Application user's responsibilities are modified:

1 Display the alert definition window from the Oracle Application Alert Manager.

2 Click the **Event** tab to define the event alert.

3 Enter **`Application Object Library`** in the Application field.

4 Enter **`FND_USER_RESP_GROUPS`** in the Table field.

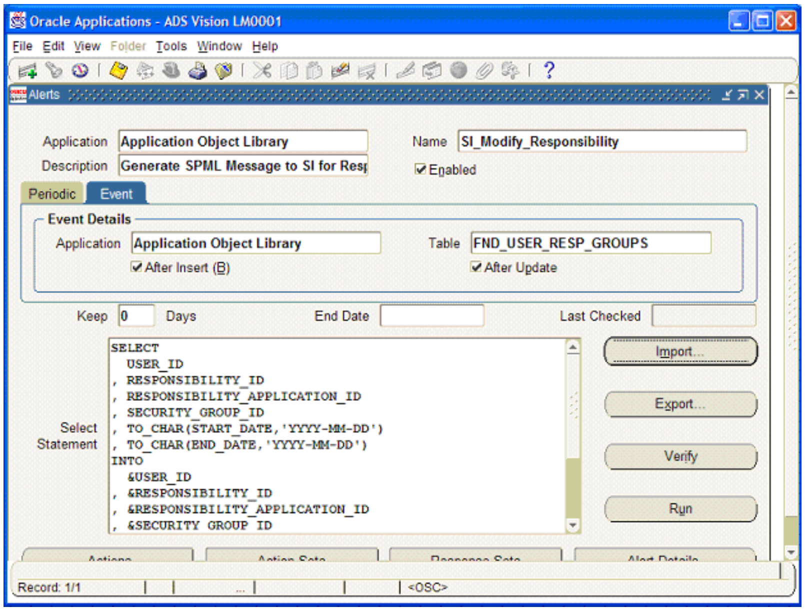
5 Select the **After Insert** and **After Update** options.

6 Enter the alert name and description.

7 Save the alert.

Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyRespSelect.sql` script.

8 Click **OK** on the file upload window.



Application	Application Object Library	Name	SI_Modify_Responsibility
Description	Generate SPML Message to SI for Resp	☑ Enabled	

Event Details

Application	Application Object Library	Table	FND_USER_RESP_GROUPS
☑ After Insert (B)		☑ After Update	

Keep	0 Days	End Date		Last Checked	

Select Statement

```
SELECT
  USER_ID
, RESPONSIBILITY_ID
, RESPONSIBILITY_APPLICATION_ID
, SECURITY_GROUP_ID
, TO_CHAR(START_DATE,'YYYY-MM-DD')
, TO_CHAR(END_DATE,'YYYY-MM-DD')
INTO
  &USER_ID
, &RESPONSIBILITY_ID
, &RESPONSIBILITY_APPLICATION_ID
, &SECURITY_GROUP_ID
```

9 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.

10 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

11 Enter the action name, such as **`SENDSPML`**. Select **Detail** as the action level, then save the alert.

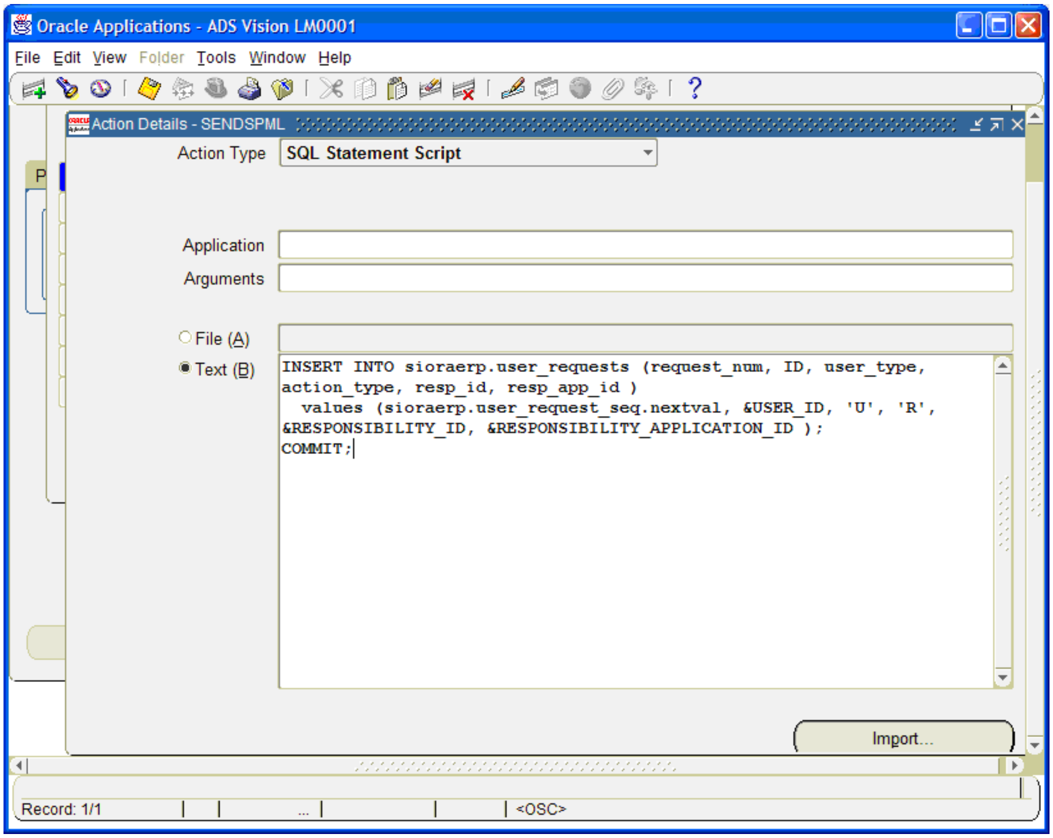
12 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

13 Select **SQL Statement Script** for the Action Type.

14 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

15 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyRespAction.sql` script. This populates the window with the content of the script.

Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Action Details - SENDSPML
Import...
Record: 1/1
<OSC>

Action Type: SQL Statement Script

Application

Arguments

○ File (A)

● Text (B)

```
INSERT INTO sioraerp.user_requests (request_num, ID, user_type,
action_type, resp_id, resp_app_id )
  values (sioraerp.user_request_seq.nextval, &USER_ID, 'U', 'R',
&RESPONSIBILITY_ID, &RESPONSIBILITY_APPLICATION_ID );
COMMIT;
```

16 Save the alert.

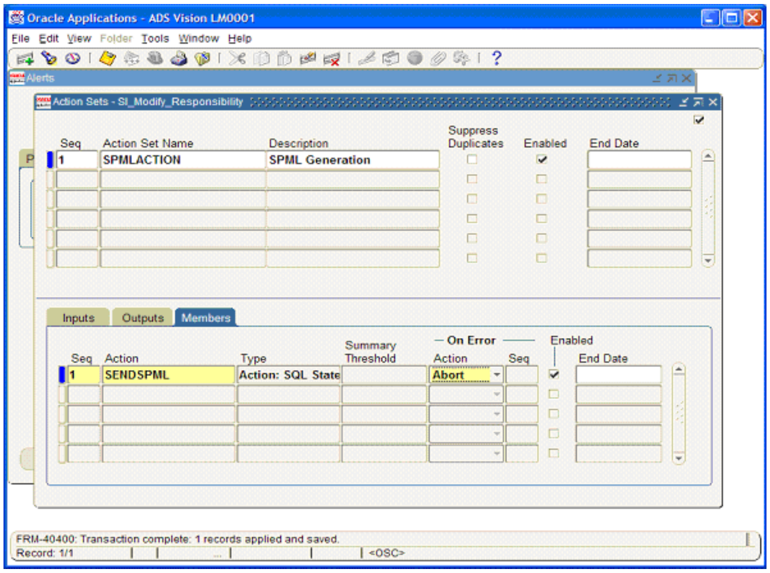
17 Close the Alert Details definition window and Action window and go to main alert definition window.

18 Click the **Action Sets** button.

19 Enter the action set name, such as **SPMLACTION**, and save.

20 Click the **Members** tab.

21 Choose the action defined above and save the alert.



Oracle Applications - ADS Vision LM0001

Action Sets - SI_Modify_Responsibility

Seq	Action Set Name	Description	Suppress Duplicates	Enabled	End Date
1	SPMLACTION	SPML Generation	□	☑	
			□	□	
			□	□	
			□	□	
			□	□	
			□	□	

Seq	Action	Type	Summary Threshold	On Error Action	On Error Seq	Enabled	End Date
1	SENDSPML	Action: SQL State		Abort		☑	
						□	
						□	
						□	
						□	

FRM-40400: Transaction complete: 1 records applied and saved.

Record: 1/1 <OSC>

22 Close the Action Sets window and click on **Alert Details** to display the Alert Details window.

23 In the alert details window, click the **Installations** tab.

24 Enter the application user ID (**`apps`**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).

25 Save the alert. The alert is now enabled and active.

Define an Alert for Application User Responsibility Modifications (v11.5.10)

Perform the following steps on Oracle 11.5.10 to define an alert that is triggered when an Application user's responsibilities are modified:

1 Display the alert definition window from the Oracle Application Alert Manager.

2 Click the **Event** tab to define the event alert.

3 Enter **`Application Object Library`** in the Application field.

4 Enter **`wf_local_user_roles`** in the Table field.

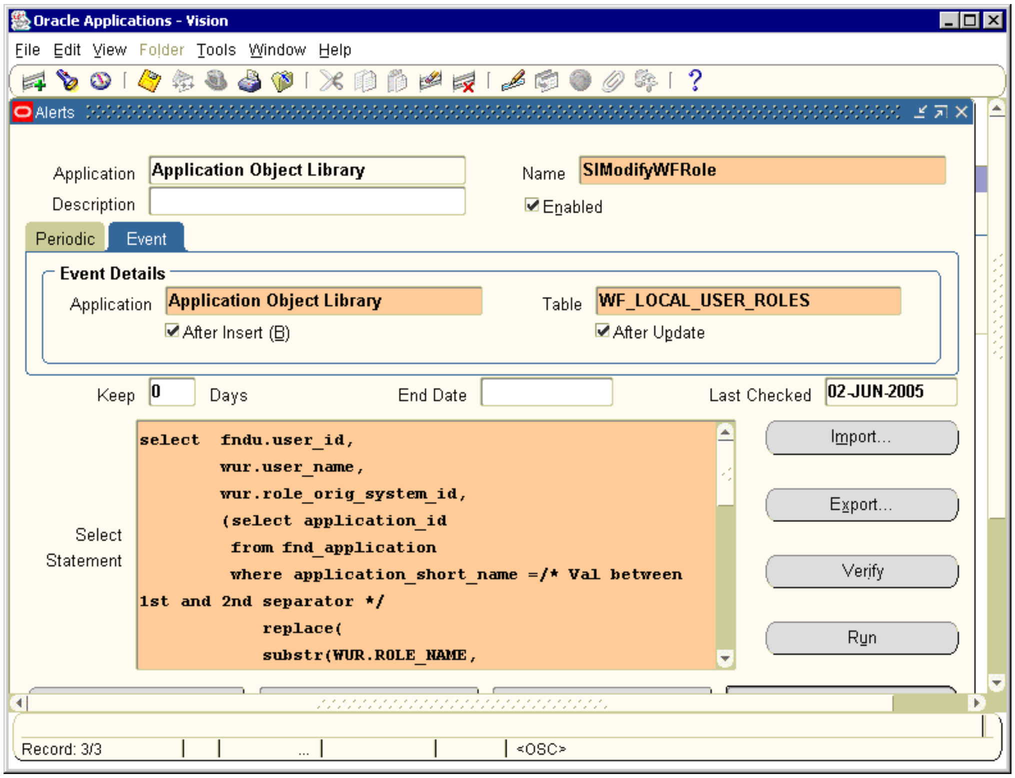
5 Select the **After Insert** and **After Update** options.

6 Enter the alert name and description.

7 Save the alert.

8 Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyWFRoleSelect.sql` script.

9 Click **OK** on the file upload window.


Oracle Applications - Vision
File Edit View Folder Tools Window Help
Alerts
Periodic
Event
Import...
Export...
Verify
Run

Application: Application Object Library

Description:

Name: SIModifyWFRole

☑ Enabled

Event Details

Application: Application Object Library

Table: WF_LOCAL_USER_ROLES

☑ After Insert (B)

☑ After Update

Keep: 0 Days

End Date:

Last Checked: 02-JUN-2005

Select Statement:

```
select  fndu.user_id,
        wur.user_name,
        wur.role_orig_system_id,
        (select application_id
         from fnd_application
         where application_short_name =/* Val between
1st and 2nd separator */
            replace(
            substr(WUR.ROLE_NAME,
```

Record: 3/3 ... <OSC>

10 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.

11 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

12 Enter the action name, such as **`SENDSPML`**. Select **Detail** as the action level, then save the alert.

13 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

14 Select **SQL Statement Script** for the Action Type.

15 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

16 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyRespAction.sql` script. This populates the window with the content of the script.

17 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

18 Save the alert.

19 Close the Alert Details definition window and Action window and go to main alert definition window.

20 Click the **Action Sets** button.

21 Enter the action set name, such as **`SPMLACTION`**, and save.

22 Click the **Members** tab.

23 Choose the action defined above and save the alert.

24 Close the Action Sets window and click **Alert Details** to display the Alert Details window.

25 In the Alert Details window, click the **Installations** tab.

26 Enter the application user ID (**apps**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).

27 Save the alert. The alert is now enabled and active.

Define an Alert for an Application User Securing Attribute Modifications

Perform the following steps to define an alert that is triggered when an Application user's Securing Attributes are modified:

1 Display the alert definition window from the Oracle Application Alert Manager.

2 Click the **Event** tab to define the event alert.

3 Enter **Oracle Common Modules-AK** in the Application field.

4 Enter **AK_WEB_USER_SEC_ATTR_VALUES** in the Table field.

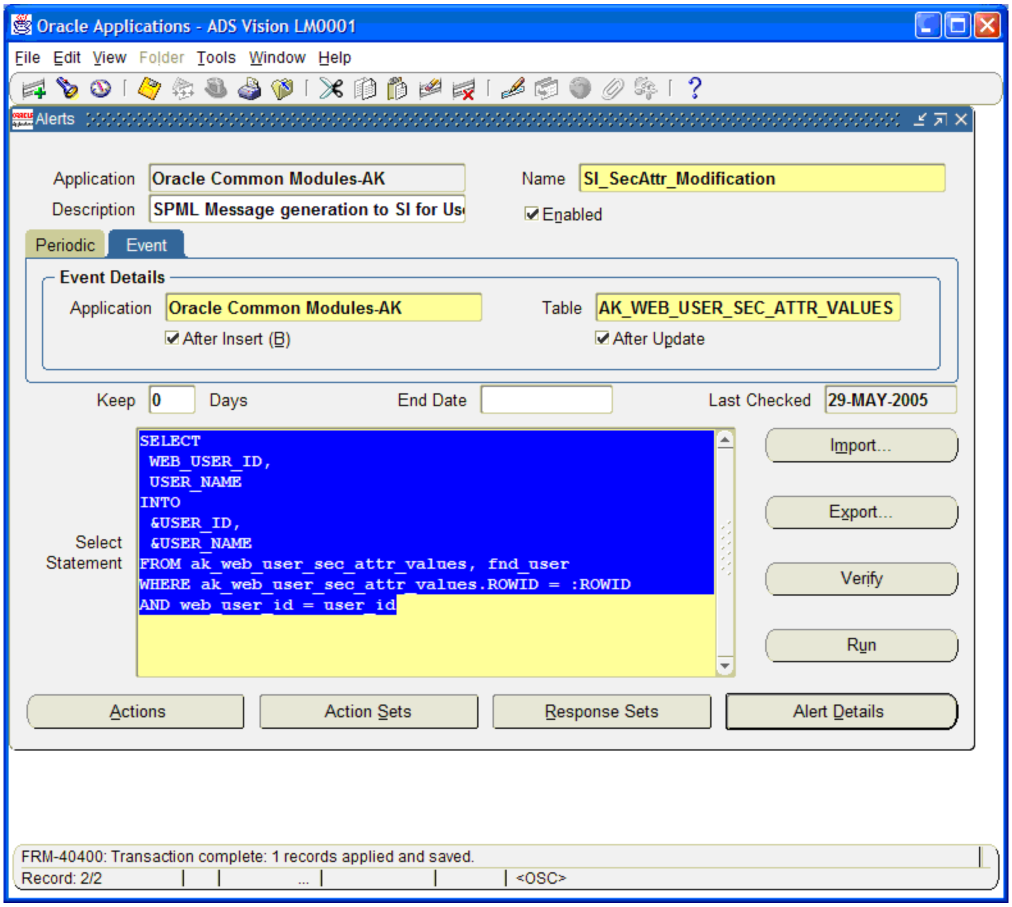
5 Select the **After Insert** and **After Update** options.

6 Enter the alert name and description.

7 Save the alert.

8 Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as C:\app\selectid on Windows) and select the ModifySecAttrSelect.sql script.

9 Click **OK** on the file upload window.



Application: Oracle Common Modules-AK

Name: SI_SecAttr_Modification

Description: SPML Message generation to SI for Us

☑ Enabled

Event Details

Application: Oracle Common Modules-AK

Table: AK_WEB_USER_SEC_ATTR_VALUES

☑ After Insert (B)

☑ After Update

Keep: 0 Days

End Date:

Last Checked: 29-MAY-2005

Select Statement:

```
SELECT
 WEB_USER_ID,
 USER_NAME
INTO
 &USER_ID,
 &USER_NAME
FROM ak_web_user_sec_attr_values, fnd_user
WHERE ak_web_user_sec_attr_values.ROWID = :ROWID
AND web_user_id = user_id
```

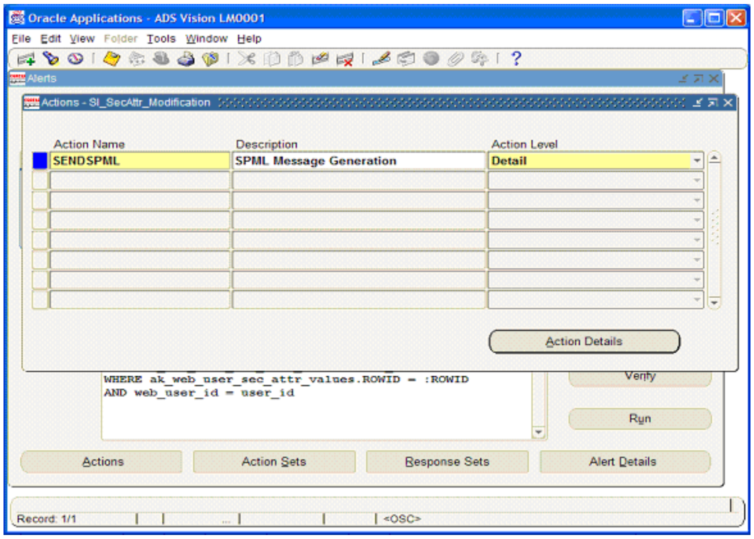
FRM-40400: Transaction complete: 1 records applied and saved.

Record: 2/2 ... <OSC>

10 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.

11 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

12 Enter the action name, such as **SENDSPML**. Select **Detail** as the action level, then save the alert.



Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_SecAttr_Modification

Action Name	Description	Action Level
SENDSPML	SPML Message Generation	Detail

```
WHERE ak_web_user_sec_attr_values.ROWID = :ROWID
AND web_user_id = user_id
```

Action Details
Verify
Run
Actions
Action Sets
Response Sets
Alert Details
Record: 1/1
<OSC>

13 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

14 Select **SQL Statement Script** for the Action Type.

15 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

16 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyUserAction.sql` script. This populates the window with the content of the script.

17 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

18 Save the alert.

19 Close the Alert Details definition window and Action window and go to main alert definition window.

20 Click the **Action Sets** button.

21 Enter the action set name, such as **SPMLACTION**, and save.

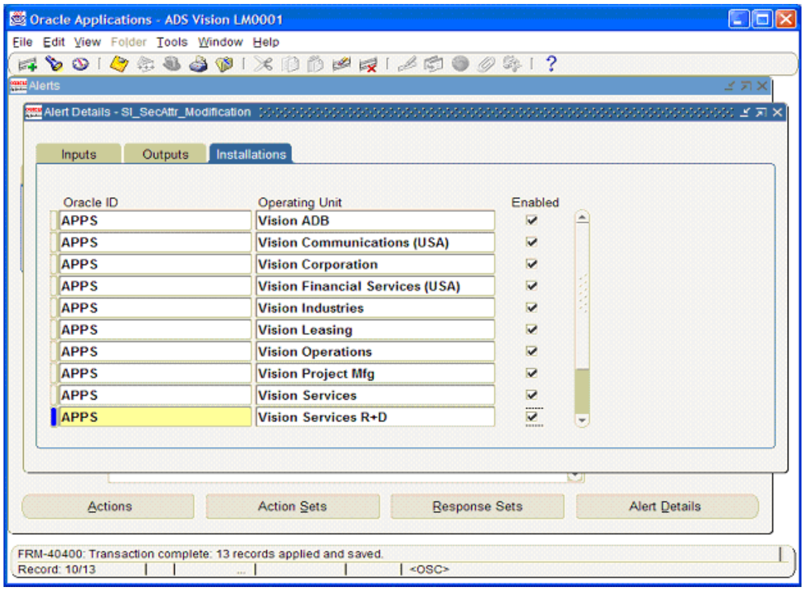
22 Click the **Members** tab.

23 Choose the action defined above and save the alert.

24 Close the Action Sets window and click on **Alert Details** to display the Alert Details window.

25 In the Alert Details window, click the **Installations** tab.

Enter the application user ID (**apps**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Alert Details - SI_SecAttr_Modification
Inputs
Outputs
Installations
Actions
Action Sets
Response Sets
Alert Details

Oracle ID	Operating Unit	Enabled
APPS	Vision ADB	☑
APPS	Vision Communications (USA)	☑
APPS	Vision Corporation	☑
APPS	Vision Financial Services (USA)	☑
APPS	Vision Industries	☑
APPS	Vision Leasing	☑
APPS	Vision Operations	☑
APPS	Vision Project Mfg	☑
APPS	Vision Services	☑
APPS	Vision Services R+D	☑

```
FRM-40400: Transaction complete: 13 records applied and saved.
Record: 10/13 | | ... | | | <OSC>
```

26 Save the alert. The alert is now enabled and active.

Create Alerts for Employee Reverse Synchronization

An Oracle Human Resources employee is stored in the Oracle Human Resources system to represent an employee in the organization. An employee is, therefore, an managed entity, not a real user on an Oracle Application system. However, an Oracle Human Resources employee can be an Oracle Application user, and vice versa.

The agent monitors the following events. If any of these events occur, the agent send the new data to Select Identity (reverse synchronization). Oracle Human Resources employees are not assigned user names; thus, the agent creates an default user name by combining the first three characters of the last name with the person ID for each employee. To change process for generating the user name, you must modify the alert select and process scripts.

- Creation of a new employee
- Changes in employee attributes
- Termination of an employee

The following employee attributes are synchronized with Select Identity:

- CompanyName
- Email
- First Name
- Last Name
- Work Phone
- Job
- Address1
- Address2
- City
- State
- Zipcode
- Country

- Position
- Grade
- Location
- Home Phone
- Date of Birth
- Manager Flag (indicates whether the employee is a manager)

Scripts are provided by the agent for the following events:

- **Add New Employee**
  Application Name: Oracle Human Resources
  Table Name: PER_PERIODS_OF_SERVICE
  Event to Monitor: After Insert
  Select SQL Script: AddEmployeeSelect.sql
  Action SQL Script: AddEmployeeAction.sql
- **Modify Employee Info**
  Application Name: Oracle Human Resources
  Table Name: PER_ALL_PEOPLE_F
  Event to Monitor: After Update
  Select SQL Script: Modify_EMP_PAPF_Select.sql
  Action SQL Script: ModifyEmployeeAction.sql
- **Modify Employee Address**
  Application Name: Oracle Human Resources
  Table Name: PER_ADDRESSES
  Event to Monitor: After Insert After Update
  Select SQL Script: Modify_EMP_ADDR_Select.sql
  Action SQL Script: ModifyEmployeeAction.sql
- **Modify Employee Job**
  Application Name: Oracle Human Resources
  Table Name: PER_ALL_ASSIGNMENTS_F
  Event to Monitor: After Insert After Update
  Select SQL Script: Modify_EMP_ASGN_Select.sql
  Action SQL Script: ModifyEmployeeAction.sql
- **Terminate Employee**
  Application Name: Oracle Human Resources
  Table Name: PER_ALL_PEOPLE_F
  Event to Monitor: After Insert
  Select SQL Script: TerminateEmployeeSelect.sql
  Action SQL Script: TerminateEmployeeAction.sq

The following sections describe how to create an alert in the Oracle Application system for each of these events.

Define a New Employee Alert

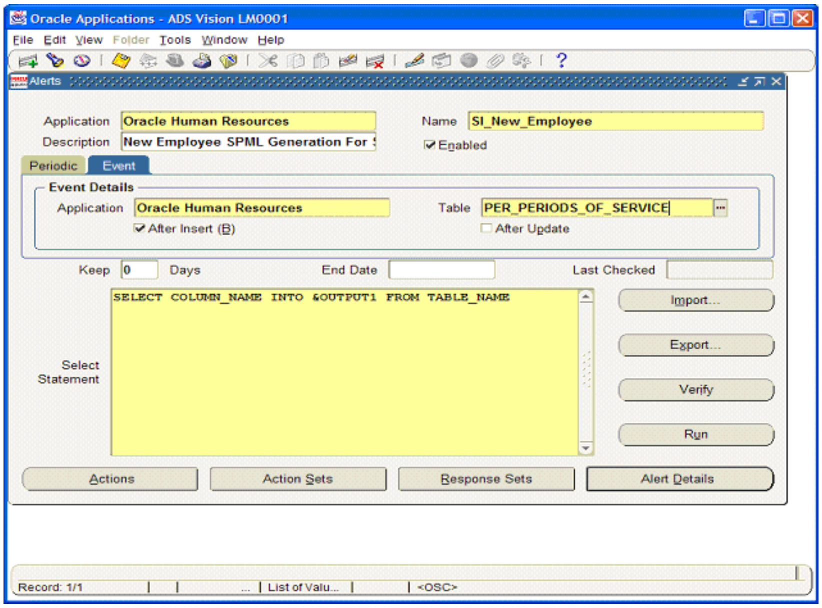
Complete the following steps to define an alert that is triggered when a new employee is created in the Oracle Human Resources system:

1. Display the alert definition window from the Oracle Application Alert Manager.
2. Click the **Event** tab to define the event alert.
3. Enter **`Oracle Human Resources`** in the Application field.
4. Enter **`PER_PERIODS_OF_SERVICE`** in the Table field.

5 Select the **After Insert** option. Make sure the **After Update** option is deselected.

6 Enter the alert name and description.

7 Save the alert.



Application: Oracle Human Resources
Name: SI_New_Employee
Description: New Employee SPML Generation For
☑ Enabled

Application: Oracle Human Resources
Table: PER_PERIODS_OF_SERVICE
☑ After Insert (B)
□ After Update

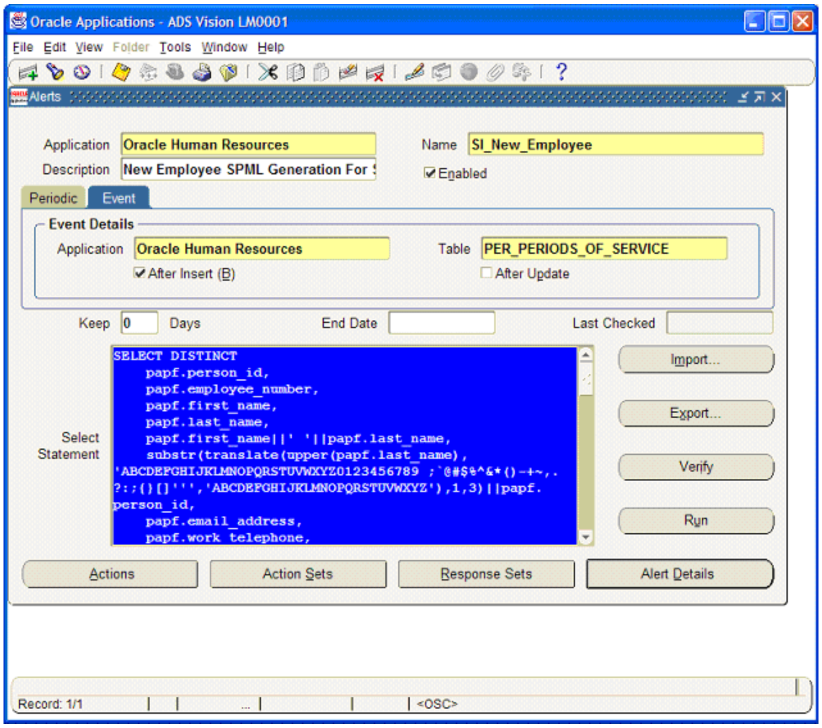
Keep: 0 Days
End Date:
Last Checked:

Select Statement:

```
SELECT COLUMN_NAME INTO &OUTPUT1 FROM TABLE_NAME
```

8 Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `AddEmployeeSelect.sql` script.

9 Click **OK** on the file upload window.



Application: Oracle Human Resources
Name: SI_New_Employee
Description: New Employee SPML Generation For
☑ Enabled

Application: Oracle Human Resources
Table: PER_PERIODS_OF_SERVICE
☑ After Insert (B)
□ After Update

Keep: 0 Days
End Date:
Last Checked:

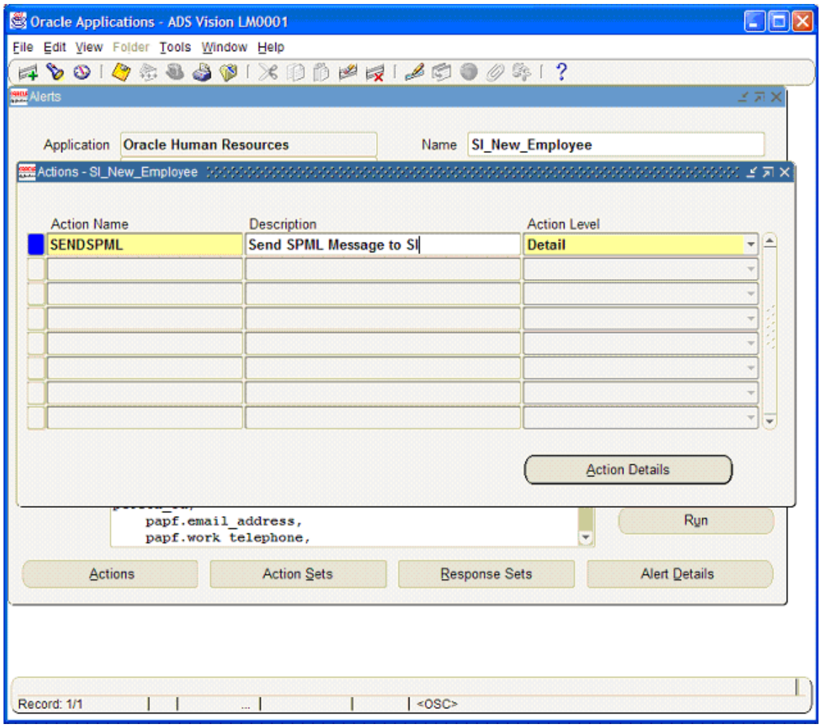
Select Statement:

```
SELECT DISTINCT
    papf.person_id,
    papf.employee_number,
    papf.first_name,
    papf.last_name,
    papf.first_name||' '||papf.last_name,
    substr(translate(upper(papf.last_name),
'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ;`@#$%^&*()-+~,.
?:;{}[]''','ABCDEFGHIJKLMNOPQRSTUVWXYZ'),1,3)||papf.
person_id,
    papf.email_address,
    papf.work_telephone,
```

10 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.

11 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

Enter the action name, such as **`SENDSPML`**. Select **Detail** as the action level, then save the alert.



Oracle Applications - ADS Vision LM0001

Alerts

Application: Oracle Human Resources

Name: SI_New_Employee

Actions - SI_New_Employee

Action Name	Description	Action Level
SENDSPML	Send SPML Message to SI	Detail

```
    papf.email_address,
    papf.work telephone,
```

Record: 1/1

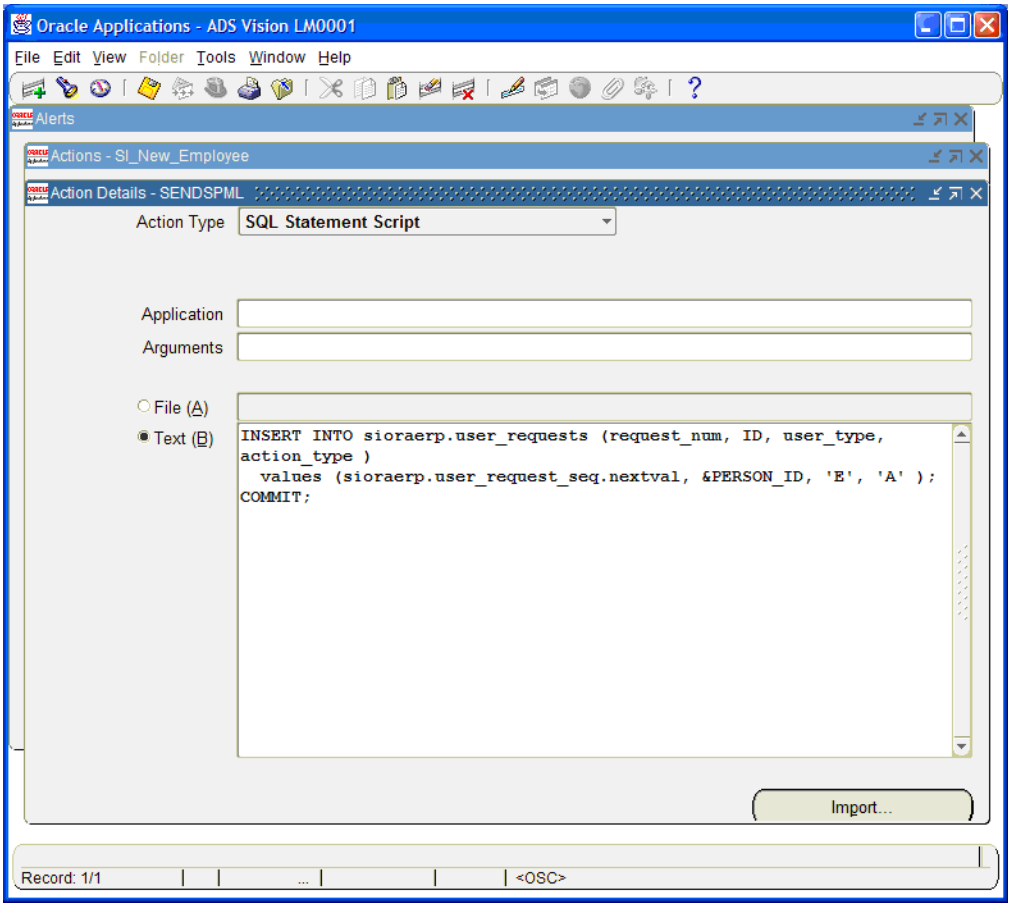
12 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

13 Select **SQL Statement Script** for the Action Type.

14 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

15 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select

the `AddEmployeeAction.sql` script. This populates the window with the content of the script.



Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_New_Employee
Action Details - SENDSPML

Action Type: SQL Statement Script

Application

Arguments

○ File (A)

● Text (B)

```
INSERT INTO sioraerp.user_requests (request_num, ID, user_type,
action_type )
  values (sioraerp.user_request_seq.nextval, &PERSON_ID, 'E', 'A' );
COMMIT;
```

Import...
Record: 1/1
...
<OSC>

16 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

17 Save the alert.

18 Close the Alert Details definition window and Action window and go to main alert definition window.

19 Click the **Action Sets** button.

20 Enter the action set name, such as **`SPMLACTION`**, and save.

21 Click the **Members** tab.

22 Choose the action defined above and save the alert.

23 Close the Action Sets window and click **Alert Details** to display the Alert Details window.

24 In the Alert Details window, click the **Installations** tab.

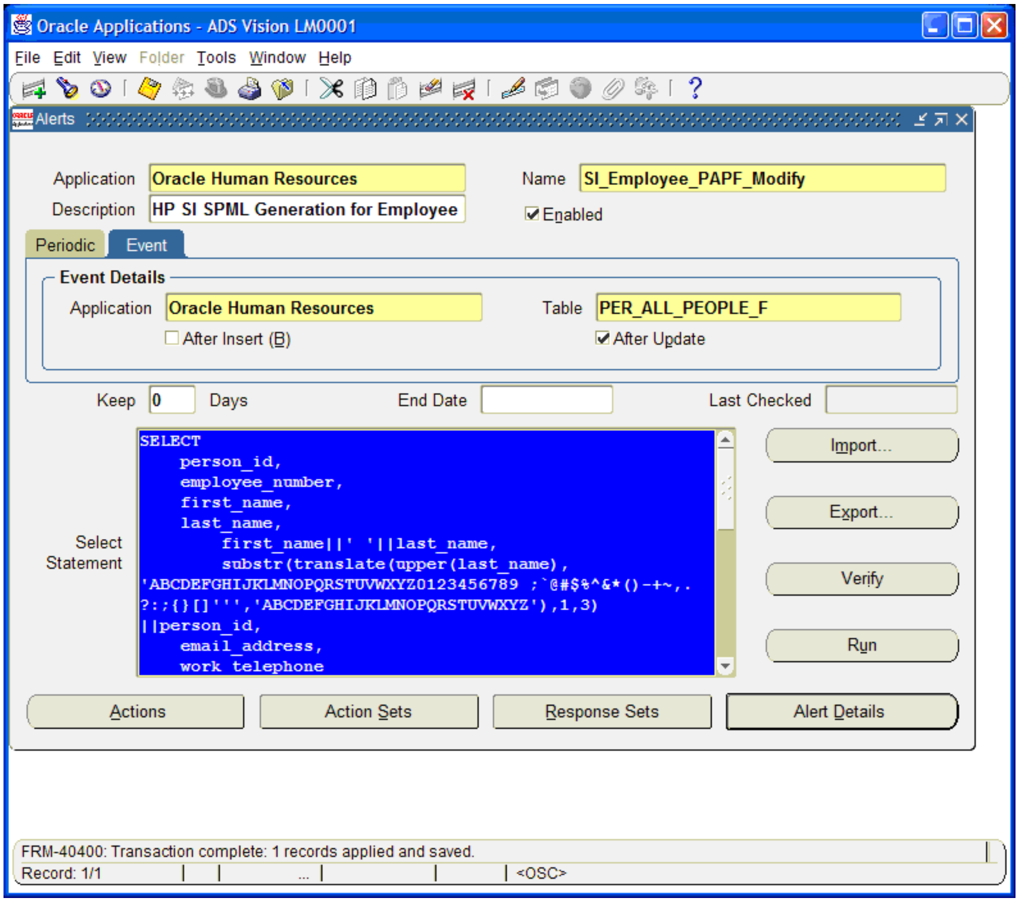
25 Enter the application user ID (**`apps`**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).

26 Save the alert. The alert is now enabled and active.

Define an Alert for Employee Modifications

Complete the following steps to define an alert that is triggered when an employee is modified in the Oracle Human Resources system:

1. Display the alert definition window from the Oracle Application Alert Manager.
2. Click the **Event** tab to define the event alert.
3. Enter **`Oracle Human Resources`** in the Application field.
4. Enter **`PER_ALL_PEOPLE_F`** in the Table field.
5. Select the **After Update** option. Make sure the **After Insert** option is deselected.
6. Enter the alert name and description.
7. Save the alert.
8. Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyEMP_PAPF_Select.sql` script.
9. Click **OK** on the file upload window.



Application: Oracle Human Resources
Name: SI_Employee_PAPF_Modify
Description: HP SI SPML Generation for Employee
☑ Enabled

Event Details
Application: Oracle Human Resources
Table: PER_ALL_PEOPLE_F
☐ After Insert (B)
☑ After Update

Keep: 0 Days
End Date:
Last Checked:

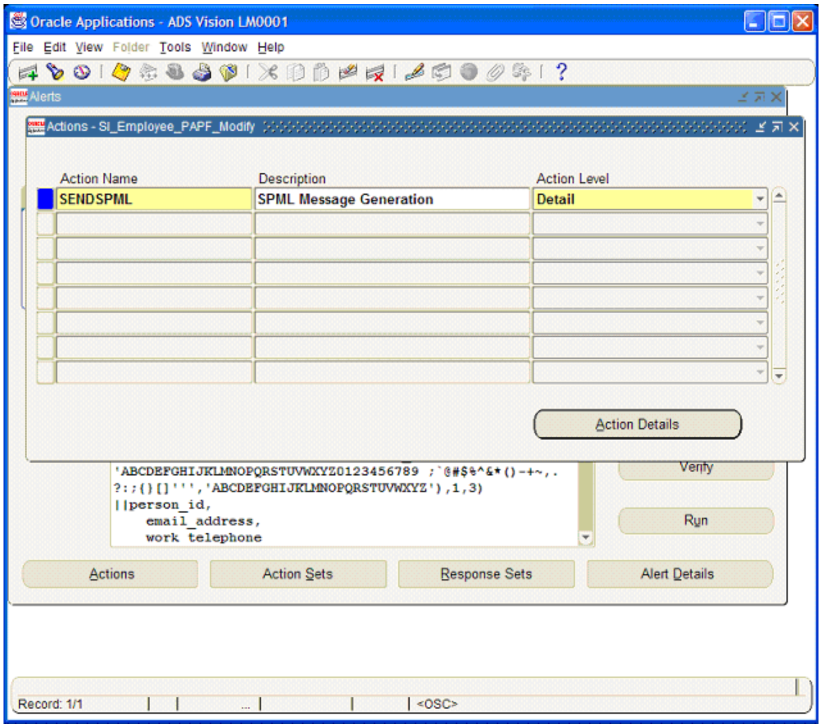
Select Statement

```
SELECT
    person_id,
    employee_number,
    first_name,
    last_name,
        first_name||' '||last_name,
        substr(translate(upper(last_name),
'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ;`@#$%^&*()-+~,.
?:;{}[]''','ABCDEFGHIJKLMNOPQRSTUVWXYZ'),1,3)
||person_id,
    email_address,
    work telephone
```

FRM-40400: Transaction complete: 1 records applied and saved.
Record: 1/1 | | ... | | | <OSC>

10. Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.
11. Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

12 Enter the action name, such as **SENDSPML**. Select **Detail** as the action level, then save the alert.



Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_Employee_PAPF_Modify

Action Name	Description	Action Level
SENDSPML	SPML Message Generation	Detail

Action Details
Verify
Run
Actions
Action Sets
Response Sets
Alert Details
Record: 1/1
<OSC>

```
'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ;`@#$%^&*()-+~,.
?:;{}[]''','ABCDEFGHIJKLMNOPQRSTUVWXYZ'),1,3)
||person_id,
    email_address,
    work_telephone
```

13 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

14 Select **SQL Statement Script** for the Action Type.

15 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

16 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyEmployeeAction.sql` script. This populates the window with the content of the script.

17 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

18 Save the alert.

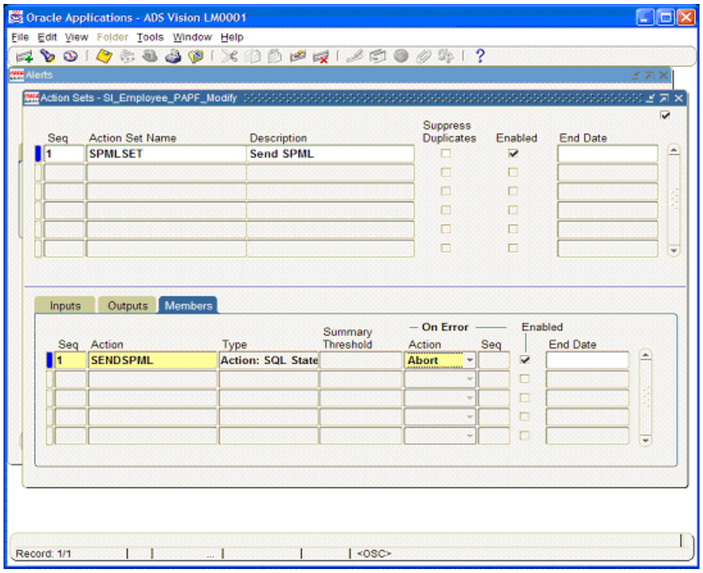
19 Close the Alert Details definition window and Action window and go to main alert definition window.

20 Click the **Action Sets** button.

21 Enter the action set name, such as **SPMLACTION**, and save.

22 Click the **Members** tab.

23 Choose the action defined above and save the alert.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Action Sets - SI_Employee_PAPF_Modify
Inputs
Outputs
Members
Record: 1/1
<OSC>

Seq	Action Set Name	Description	Suppress Duplicates	Enabled	End Date
1	SPMLSET	Send SPML	□	☑	

				On Error		Enabled	
Seq	Action	Type	Summary Threshold	Action	Seq		End Date
1	SENDSPML	Action: SQL State		Abort		☑	

24 Close the Action Sets window and click on **Alert Details** to display the Alert Details window.

25 In the alert details window, click the **Installations** tab.

26 Enter the application user ID (**`apps`**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).

27 Save the alert. The alert is now enabled and active.

Define an Alert for Employee Address Modifications

Perform the following steps to define an alert that is triggered when an employee's address is modified in the Oracle Human Resources system:

1 Display the alert definition window from the Oracle Application Alert Manager.

2 Click the **Event** tab to define the event alert.

3 Enter **`Oracle Human Resources`** in the Application field.

4 Enter **`PER_ADDRESSES`** in the Table field.

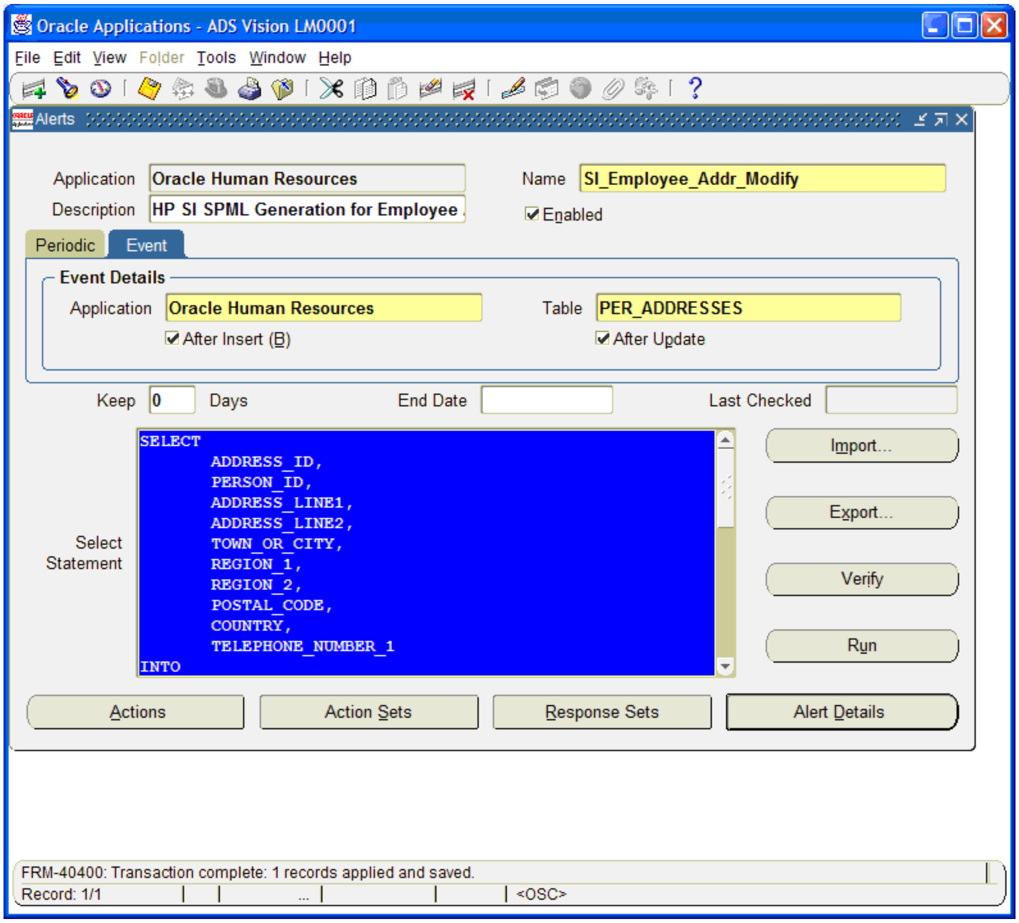
5 Select the **After Insert** and **After Update** options.

6 Enter the alert name and description.

7 Save the alert.

8 Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyEMP_ADDR_Select.sql` script.

9 Click **OK** on the file upload window.



Application	Oracle Human Resources	Name	SI_Employee_Addr_Modify
Description	HP SI SPML Generation for Employee	☑ Enabled	

Event Details

Application	Oracle Human Resources	Table	PER_ADDRESSES
☑ After Insert (B)		☑ After Update	

Keep 0 Days End Date Last Checked

Select Statement

```
SELECT
       ADDRESS_ID,
       PERSON_ID,
       ADDRESS_LINE1,
       ADDRESS_LINE2,
       TOWN_OR_CITY,
       REGION_1,
       REGION_2,
       POSTAL_CODE,
       COUNTRY,
       TELEPHONE_NUMBER_1
INTO
```

```
FRM-40400: Transaction complete: 1 records applied and saved.
Record: 1/1 | | ... | | | <OSC>
```

10 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.

11 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

12 Enter the action name, such as **SENDSPML**. Select **Detail** as the action level, then save the alert.

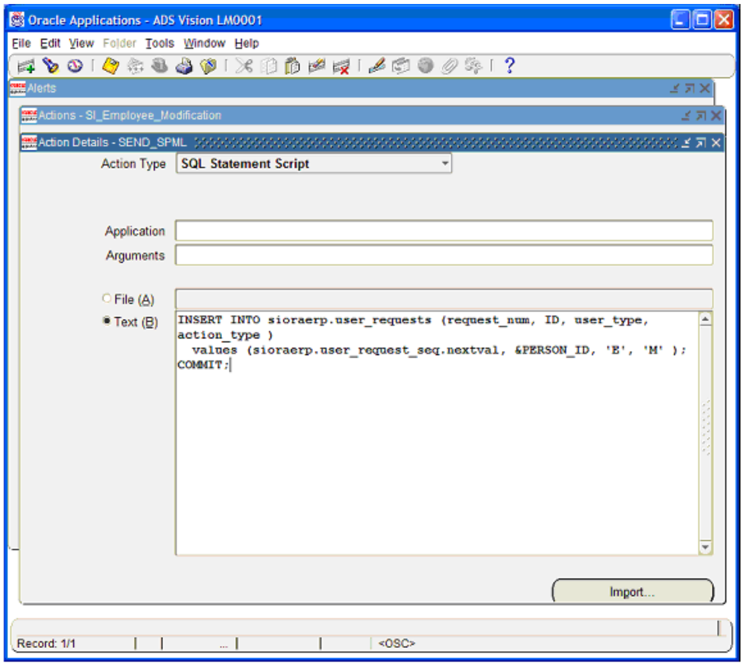
13 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

14 Select **SQL Statement Script** for the Action Type.

15 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

16 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select

the `ModifyEmployeeAction.sql` script. This populates the window with the content of the script.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_Employee_Modification
Action Details - SEND_SPML
Import...
Record: 1/1
<OSC>

Action Type: SQL Statement Script

Application:

Arguments:

○ File (A)

● Text (B)

```
INSERT INTO sioraerp.user_requests (request_num, ID, user_type,
action_type )
  values (sioraerp.user_request_seq.nextval, &PERSON_ID, 'E', 'M' );
COMMIT;
```

17 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

18 Save the alert.

19 Close the Alert Details definition window and Action window and go to main alert definition window.

20 Click the **Action Sets** button.

21 Enter the action set name, such as **`SPMLACTION`**, and save.

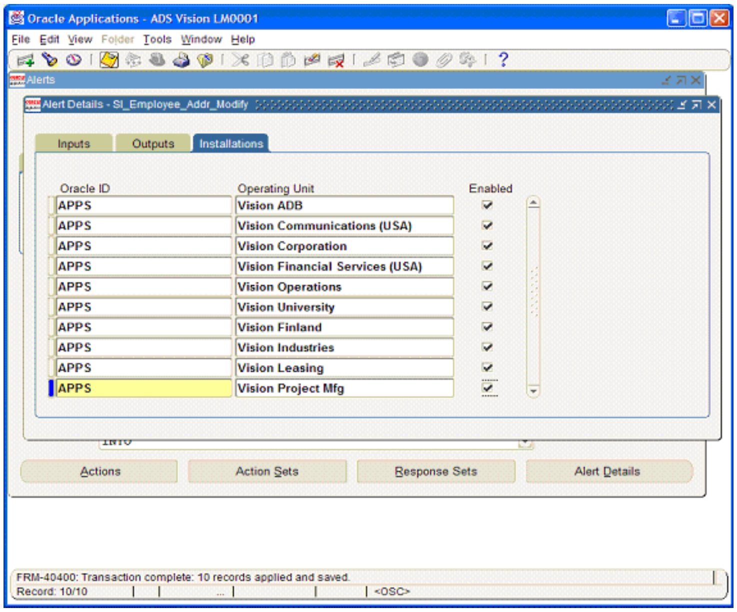
22 Click the **Members** tab.

23 Choose the action defined above and save the alert.

24 Close the Action Sets window and click on **Alert Details** to display the Alert Details window.

25 In the alert details window, click the **Installations** tab.

26 Enter the application user ID (**`apps`**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Alert Details - SI_Employee_Addr_Modify
Inputs
Outputs
Installations
INTO
Actions
Action Sets
Response Sets
Alert Details

Oracle ID	Operating Unit	Enabled
APPS	Vision ADB	☑
APPS	Vision Communications (USA)	☑
APPS	Vision Corporation	☑
APPS	Vision Financial Services (USA)	☑
APPS	Vision Operations	☑
APPS	Vision University	☑
APPS	Vision Finland	☑
APPS	Vision Industries	☑
APPS	Vision Leasing	☑
APPS	Vision Project Mfg	☑

FRM-40400: Transaction complete: 10 records applied and saved.

Record: 10/10 ... <OSC>

27 Save the alert. The alert is now enabled and active.

Define an Alert for Employee Position Modifications

Perform the following steps to define an alert that is triggered when an employee's assignments are modified in the Oracle Human Resources system:

1 Display the alert definition window from the Oracle Application Alert Manager.

2 Click the **Event** tab to define the event alert.

3 Enter **`Oracle Human Resources`** in the Application field.

4 Enter **`PER_ALL_ASSIGNMENTS_F`** in the Table field.

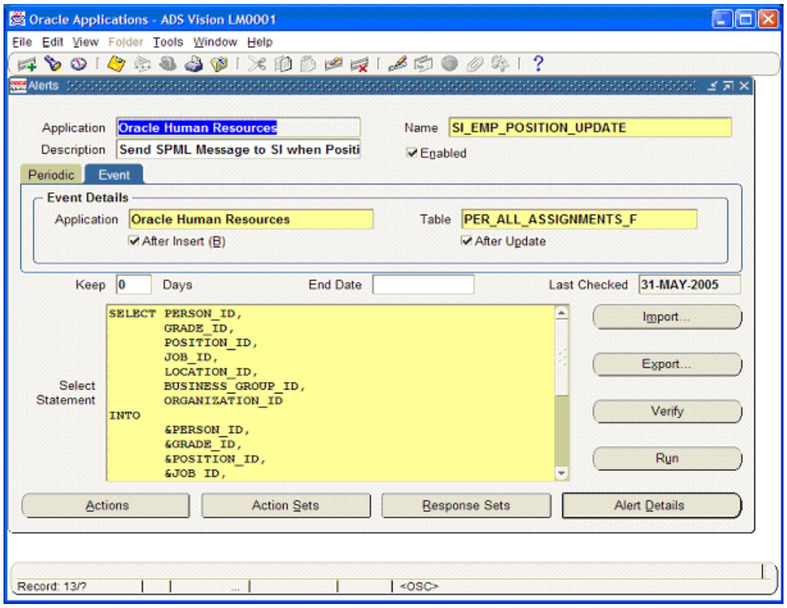
5 Select the **After Insert** and **After Update** options.

6 Enter the alert name and description.

7 Save the alert.

8 Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `ModifyEMP_ASGN_Select.sql` script.

9 Click **OK** on the file upload window.



Application: Oracle Human Resources
Description: Send SPML Message to SI when Positi
Name: SI_EMP_POSITION_UPDATE
☑ Enabled

Event Details
Application: Oracle Human Resources
Table: PER_ALL_ASSIGNMENTS_F
☑ After Insert (B)
☑ After Update

Keep: 0 Days
End Date:
Last Checked: 31-MAY-2005

Select Statement:

```
SELECT  PERSON_ID,
        GRADE_ID,
        POSITION_ID,
        JOB_ID,
        LOCATION_ID,
        BUSINESS_GROUP_ID,
        ORGANIZATION_ID
INTO
        &PERSON_ID,
        &GRADE_ID,
        &POSITION_ID,
        &JOB_ID,
```

10 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.

11 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

12 Enter the action name, such as **`SENDSPML`**. Select **Detail** as the action level, then save the alert.

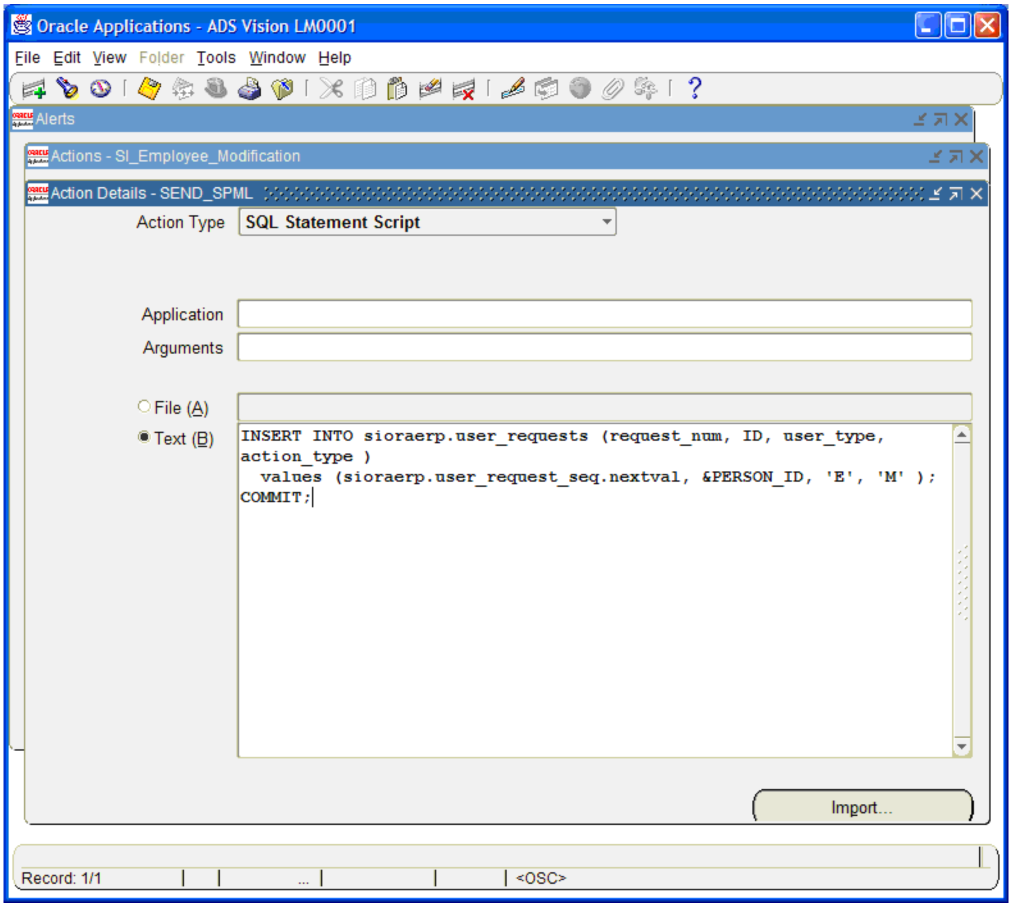
13 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

14 Select **SQL Statement Script** for the Action Type.

15 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

16 Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select

the `ModifyEmployeeAction.sql` script. This populates the window with the content of the script.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_Employee_Modification
Action Details - SEND_SPML
Application
Arguments
○ File (A)
Import...
Record: 1/1
<OSC>

Action Type: SQL Statement Script

● Text (B)

```
INSERT INTO sioraerp.user_requests (request_num, ID, user_type,
action_type )
  values (sioraerp.user_request_seq.nextval, &PERSON_ID, 'E', 'M' );
COMMIT;
```

17 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

18 Save the alert.

19 Close the Alert Details definition window and Action window and go to main alert definition window.

20 Click the **Action Sets** button.

21 Enter the action set name, such as **`SPMLACTION`**, and save.

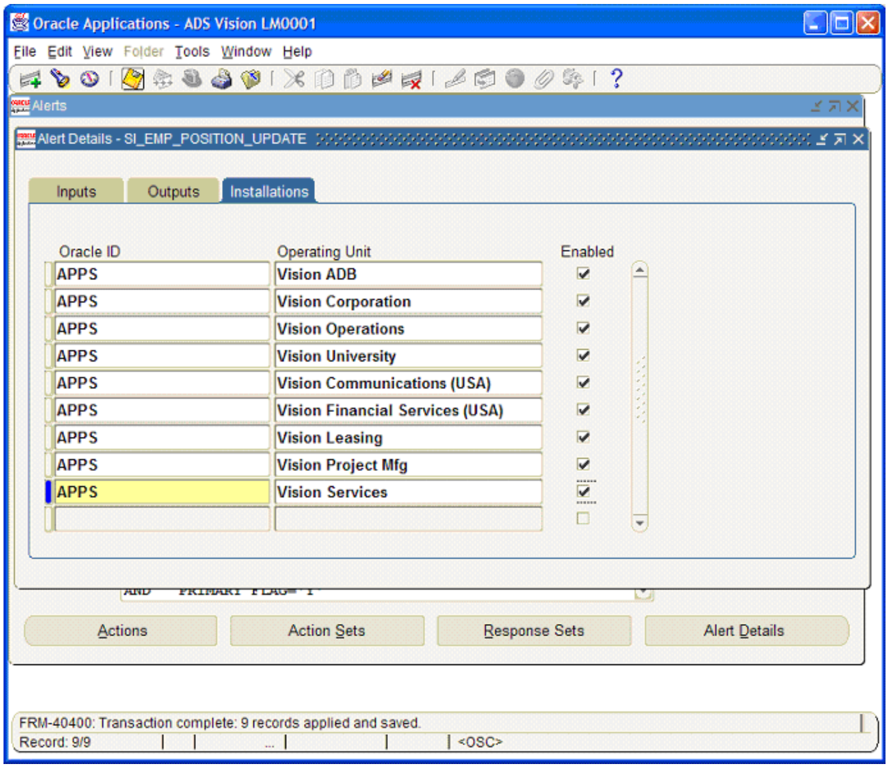
22 Click the **Members** tab.

23 Choose the action defined above and save the alert.

24 Close the Action Sets window and click on **Alert Details** to display the Alert Details window.

25 In the alert details window, click the **Installations** tab.

26 Enter the application user ID (**apps**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Alert Details - SI_EMP_POSITION_UPDATE
Inputs
Outputs
Installations
Actions
Action Sets
Response Sets
Alert Details

Oracle ID	Operating Unit	Enabled
APPS	Vision ADB	☑
APPS	Vision Corporation	☑
APPS	Vision Operations	☑
APPS	Vision University	☑
APPS	Vision Communications (USA)	☑
APPS	Vision Financial Services (USA)	☑
APPS	Vision Leasing	☑
APPS	Vision Project Mfg	☑
APPS	Vision Services	☑
		□

FRM-40400: Transaction complete: 9 records applied and saved.
Record: 9/9 <OSC>

27 Save the alert. The alert is now enabled and active.

Define an Employee Termination Alert

Complete the following steps to define an alert that is triggered when an employee is terminated in the Oracle Human Resources system:

> Oracle HRMS uses date tracking to control employee termination, and the employee termination date can be set in the future. If the employee termination date is set in the future, the alert processing script (`SPMLProcess.sql`) flags the event for future processing. The employee is terminated in Select Identity at the actual termination date specified in the employee termination window.

1 Display the alert definition window from the Oracle Application Alert Manager.

2 Click the **Event** tab to define the event alert.

3 Enter **`Oracle Human Resources`** in the Application field.

4 Enter **`PER_ALL_PEOPLE_F`** in the Table field.

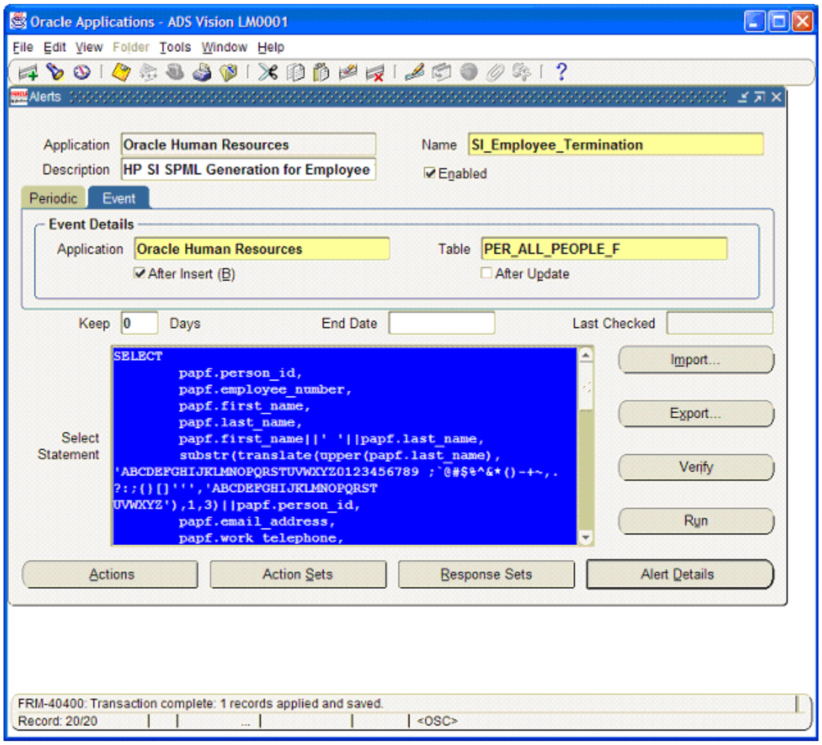
5 Select the **After Insert** option. Make sure the **After Update** option is deselected.

6 Enter the alert name and description.

7 Click **OK** on the file upload window.

8 Save the alert.

9 Click the **Import** button on the right to import the Select Statement. A file upload window is displayed. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `TerminateEmployeeSelect.sql` script.



Application: Oracle Human Resources

Description: HP SI SPML Generation for Employee

Name: SI_Employee_Termination

☑ Enabled

Event Details

Application: Oracle Human Resources

Table: PER_ALL_PEOPLE_F

☑ After Insert (B)

☐ After Update

Keep: 0 Days

End Date:

Last Checked:

Select Statement:

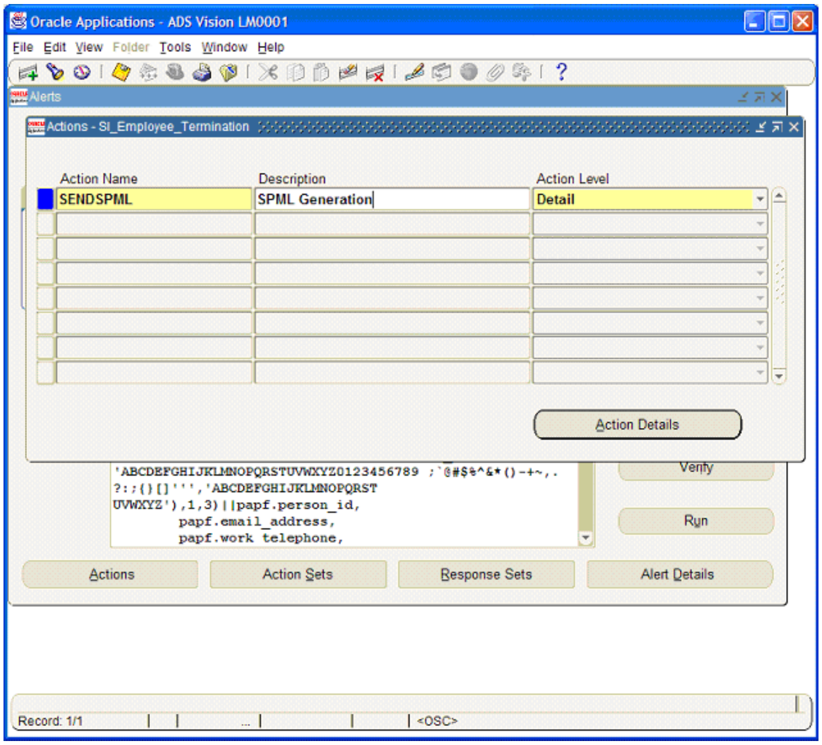
```
SELECT
        papf.person_id,
        papf.employee_number,
        papf.first_name,
        papf.last_name,
        papf.first_name||' '||papf.last_name,
        substr(translate(upper(papf.last_name),
'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ;`@#$%^&*()-+~,.
?:;{}[]''','ABCDEFGHIJKLMNOPQRST
UVWXYZ'),1,3)||papf.person_id,
        papf.email_address,
        papf.work_telephone,
```

```
FRM-40400: Transaction complete: 1 records applied and saved.
Record: 20/20 | | ... | | | <OSC>
```

10 Verify the statement by clicking the **Verify** button on the Select Statement window, which is populated with the SQL script from the imported file.

11 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

12 Enter the action name, such as **`SENDSPML`**. Select **Detail** as the action level, then save the alert.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_Employee_Termination
Action Details
Verify
Run
Actions
Action Sets
Response Sets
Alert Details
Record: 1/1
<OSC>

Action Name	Description	Action Level
SENDSPML	SPML Generation	Detail

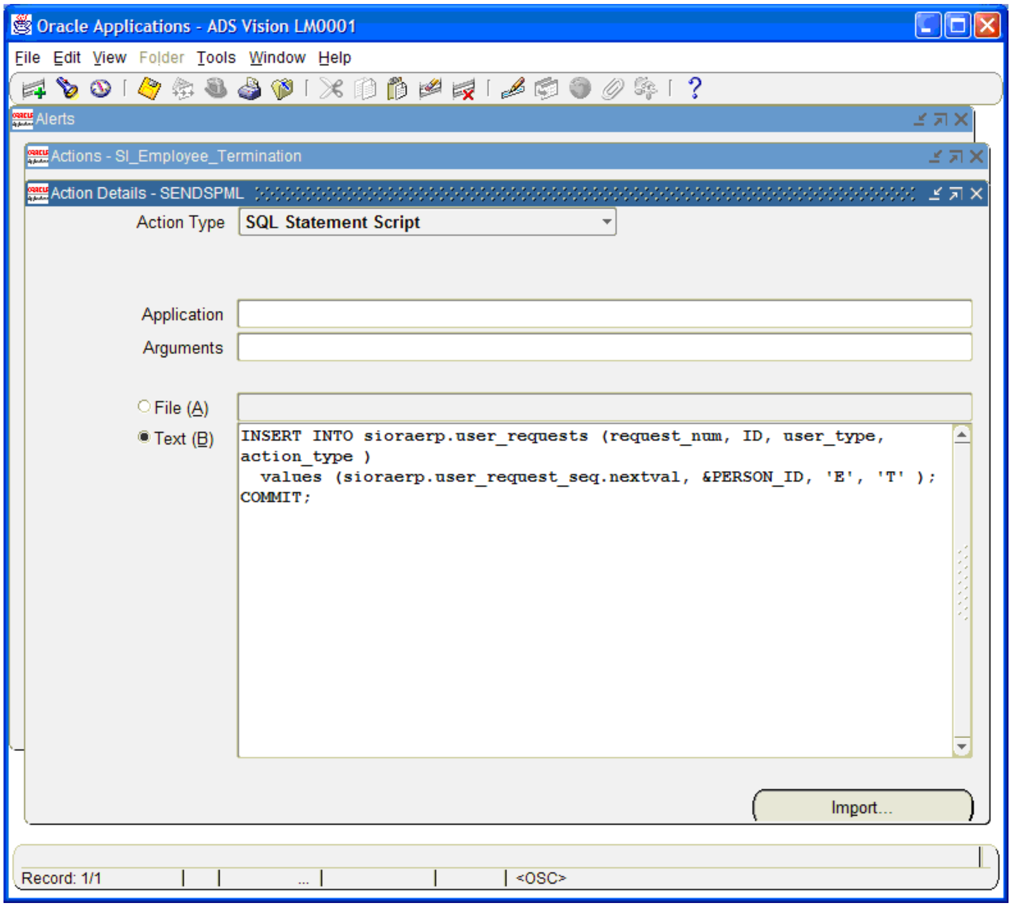
```
'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ;`@#$%^&*()-+~,.
?:;{}[]''','ABCDEFGHIJKLMNOPQRST
UVWXYZ'),1,3)||papf.person_id,
         papf.email_address,
         papf.work telephone,
```

13 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

14 Select **SQL Statement Script** for the Action Type.

15 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

Click the **Import** button to import the SQL action script. Browse to the Select Identity directory created in step 2 on page 17 (such as `C:\app\selectid` on Windows) and select the `TerminateEmployeeAction.sql` script. This populates the window with the content of the script.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_Employee_Termination
Action Details - SENDSPML
Import...
Record: 1/1
<OSC>

Action Type: SQL Statement Script

Application:

Arguments:

○ File (A)

● Text (B)

```
INSERT INTO sioraerp.user_requests (request_num, ID, user_type,
action_type )
  values (sioraerp.user_request_seq.nextval, &PERSON_ID, 'E', 'T' );
COMMIT;
```

16 Edit the script and delete the control characters at the end of each line. You may have to retype the last character of each line.

17 Save the alert.

18 Close the Alert Details definition window and Action window and go to main alert definition window.

19 Click the **Action Sets** button.

20 Enter the action set name, such as **SPMLACTION**, and save.

21 Click the **Members** tab.

22 Choose the action defined above and save the alert.

23 Close the Action Sets window and click on **Alert Details** to display the Alert Details window.

24 In the alert details window, click the **Installations** tab.

25 Enter the application user ID (**apps**) and the operating units (in case of multi-orgs) where the alert will be activated. It is important to specify all the operating units that the alert may be active (such as Vision Corporation, Vision Operations, and so on).

26 Save the alert. The alert is now enabled and active.

The SPMLProcess.sql script and SPML Messages

The event alerts insert action records into the SIORAERP.USER_REQUESTS table every time a user activity occurs. At regular intervals, the `SPMLProcess.sql` script can be run to process the individual user requests that were collected. `SPMLProcess.sql` performs following actions:

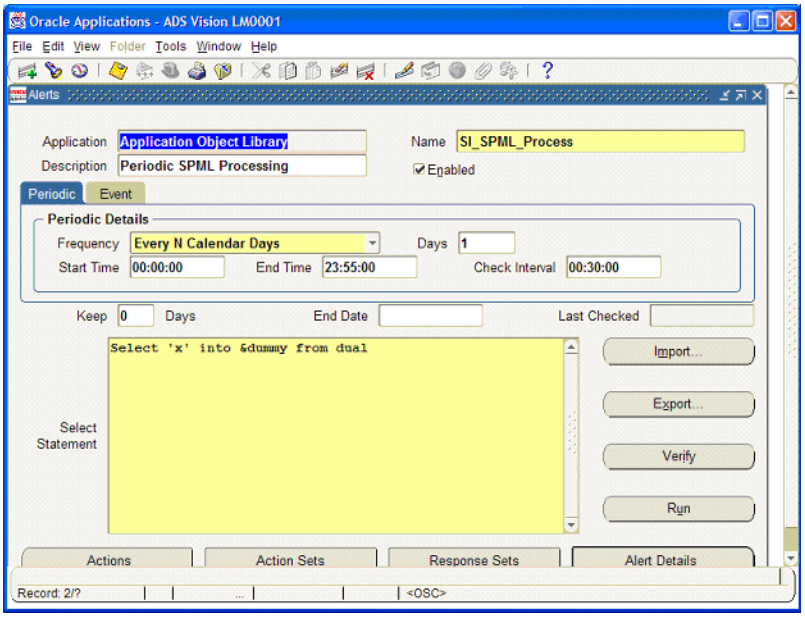
- Consolidates the user requests into provision_requests for each user. For example, multiple modify requests can be triggered for a user over time. The consolidation combines all the events into a single modify request.
- Performs retry processing. If an SPML message failed during processing, the MAXRETRY parameter in `SIConfig.sql` file specifies how many times SPML message is resent. The RETRYINTVALMIN specifies the number of minutes to wait before retry processing.
- Performs hung process processing. Resource-side errors can terminate the script before completing the SPML process. The parameter HUNGINTVALMIN specifies the number of minutes to wait before reprocessing the request.
- Processes the SPML message. Based on the user type, the `SPMLProcess.sql` script sends the appropriate attributes with add, modify, and terminate requests.

You can define a periodic alert, a concurrent program, or a job scheduler to process SPML messages. The following section describe how to define these.

Defining a Periodic Alert

A periodic alert can be defined to run the `SPMLProcess.sql` script at regular intervals. To define the period alert, complete these steps:

1. Display the alert definition window from the Oracle Application Alert Manager.
2. Click the **Periodic** tab to define the periodic alert.
3. Enter **`Application Object Library`** in the Application field.
4. Enter the alert name and description.
5. Save the alert.
6. Enter **`Select 'x' into &dummy from dual`** in the Select statement window and verify the statement.
7. Select **Every Day** as the frequency.
8. Enter **`00:00:00`** as the start time and **`23:55:00`** as the end time.
9. Enter **`00:30:00`** (every 30 minutes) as the check interval.



Application: Application Object Library
Description: Periodic SPML Processing
Name: SI_SPML_Process
☑ Enabled

Frequency: Every N Calendar Days
Days: 1
Start Time: 00:00:00
End Time: 23:55:00
Check Interval: 00:30:00

Keep: 0 Days
End Date:
Last Checked:

Select Statement:

```
select 'x' into &dummy from dual
```

10 Define the action script by clicking the **Actions** button on the bottom left-hand side of the window.

11 Enter the action name, such as **`SENDSPML`**. Select **Detail** as the action level, then save the alert.

12 Click the **Action Details** button while the action is highlighted. The Action Detail definition window is displayed.

13 Select **SQL Statement Script** for the Action Type.

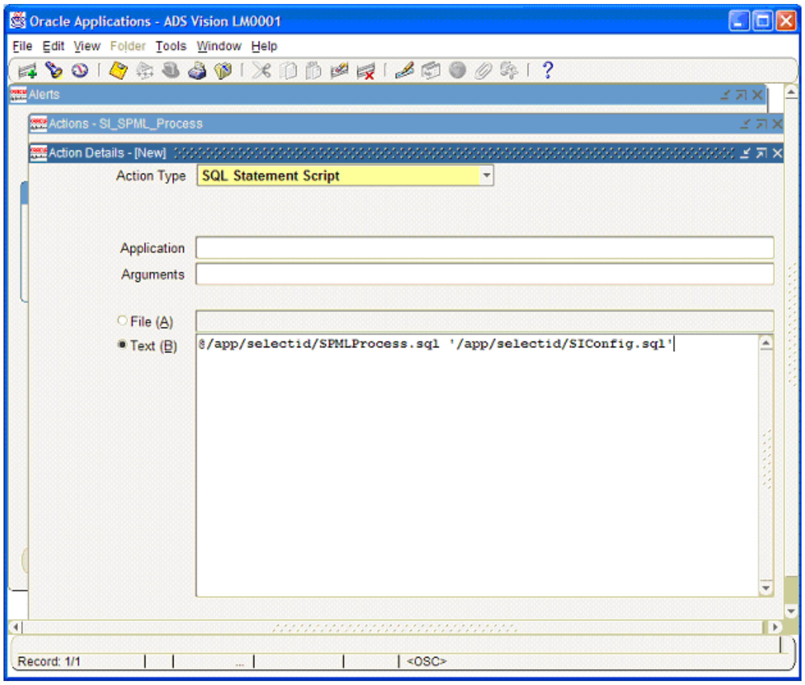
14 Click **Text** to define the SQL statement in the window. Make sure that the Application and Arguments fields are empty.

15 Enter **`@`*`SIAgentInstallDir`*`/SPMLProcess.sql '`*`SIAgentInstallDir`*`/SIConfig.sql'`** in the text field. For example, for Windows, if the Select Identity installation directory is `C:\app\selectid`, enter the following:

```
@c:\app\selectid\SPMLProcess.sql  'c:\app\selectid\SIConfig.sql'
```

Enter the following for UNIX:

```
@/app/selectid/SPMLProcess.sql '/app/selectid/SIConfig.sql'
```



Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Actions - SI_SPML_Process
Action Details - [New]
Record: 1/1
<OSC>

Action Type: SQL Statement Script

Application:

Arguments:

○ File (A)

● Text (B)

```
@/app/selectid/SPMLProcess.sql '/app/selectid/SIConfig.sql'
```

16 Save the alert.

17 Close the Alert Details definition window and Action window and go to main alert definition window.

18 Click the **Action Sets** button.

19 Enter the action set name, such as **SPMLACTION**, and save.

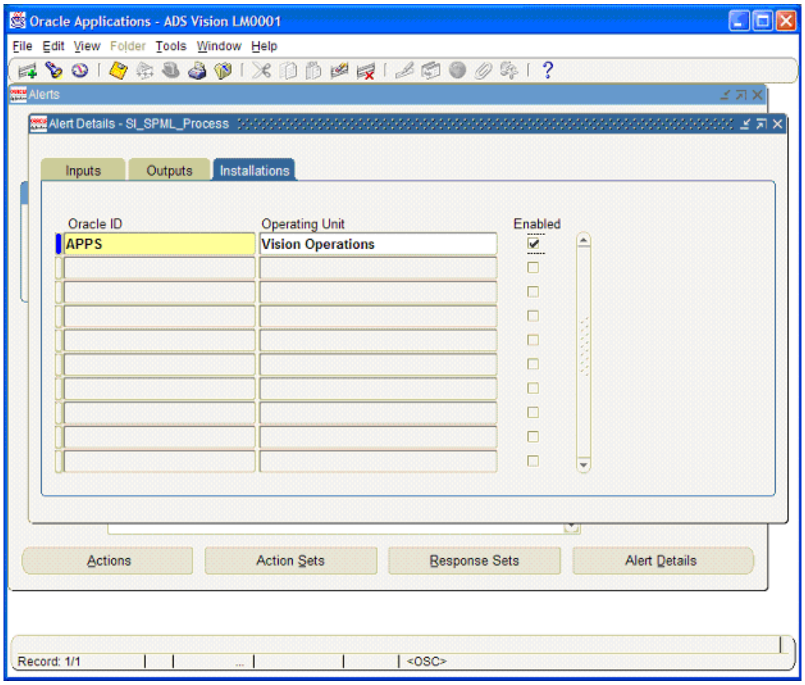
20 Click the **Members** tab.

21 Choose the action defined above and save the alert.

22 Close the Action Sets window and click on **Alert Details** to display the Alert Details window.

23 In the alert details window, click the **Installations** tab.

24 Enter the application user ID (**apps**) and the operating units (in case of multi-orgs) where the alert will be activated. For each installation, specify that the periodic alert will process `SPMLProcess.sql` again, and define the main installation for the sysadmin user only.


Oracle Applications - ADS Vision LM0001
File Edit View Folder Tools Window Help
Alerts
Alert Details - SI_SPML_Process
Inputs
Outputs
Installations
Actions
Action Sets
Response Sets
Alert Details
Record: 1/1
<OSC>

Oracle ID	Operating Unit	Enabled
APPS	Vision Operations	☑
		□
		□
		□
		□
		□
		□
		□
		□
		□

25 Make sure periodic alert scheduler is enabled in the Request → Schedule form from the Alert Manager menu.

26 Save the alert. The alert is now enabled and active.

27 Make sure the periodic alert is scheduled in the Request → Schedule window.

Define a Concurrent Program

Instead of defining a periodic alert, you can define a concurrent request to process the SPML messages. A concurrent request provides an additional level of control and a printing option. To define a concurrent program, complete the following steps:

1 Modify the `sispml.sql` script to include the correct path to the agent script location. Here is an example for Windows:

```
@c:\\app\\selectid\\SPMLProcess.sql 'c:\\app\\selectid\\SIConfig.sql'
```

Here is an example for UNIX:

```
@/app/selectid/SPMLProcess.sql '/app/selectid/SIConfig.sql'
```

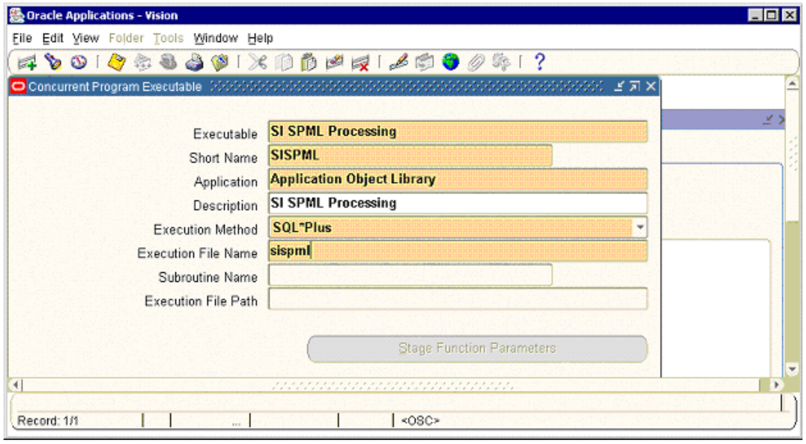
2 Copy the `sispml.sql` script to the `$FND_TOP/sql` (on UNIX) or `%FND_TOP%\sql` (on Windows) directory on the application server. Ensure that the file is readable by all. To set the environment variable, run the environment setup script in the Oracle Application home directory, as in these examples:

`/app/oracle/visappl/VIS_alimosa.env` (on UNIX)

`f:\oracle\visappl\VIS_kod.cmd` (on Windows)

3 Log on to Oracle Applications as SYSADMIN.

4. Select **Concurrent** → **Program** → **Executables**.
5. Enter the executable and description of the custom program.
6. Enter **`SISPML`** as the short name for the custom program.
7. Enter **`Application Object Library`** in the Application field.
8. Select **SQL*Plus** for the execution method.
9. Enter **`sispml`** as the execution file name. This is the script copied in step 2.
10. Save the executable definition.

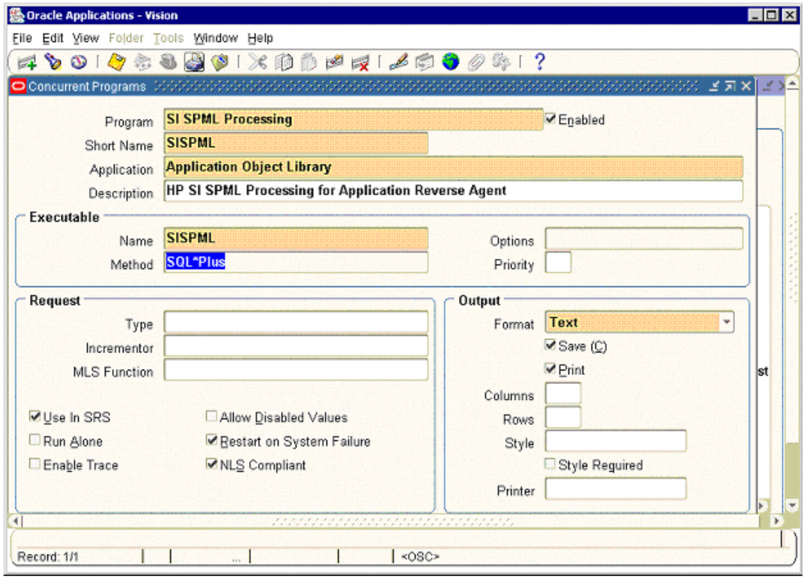

Oracle Applications - Vision
File Edit View Folder Tools Window Help
Concurrent Program Executable
Stage Function Parameters

Executable	SI SPML Processing
Short Name	SISPML
Application	Application Object Library
Description	SI SPML Processing
Execution Method	SQL*Plus
Execution File Name	sispml
Subroutine Name	
Execution File Path	

Record: 1/1 <OSC>

11. Select **Concurrent** → **Program** → **Define**.
12. Enter a program name and description.
13. Enter **`SISPML`** for the short name.
14. Enter **`Application Object Library`** in the Application field.
15. Enter the **`SISPML`** as the executable name. This is the short name defined above. If different short name is used, enter that name.
16. Select the **User in SRS**, **Restart on System Failure**, **NLS Compliant**, **Save**, and **Print** options.

17 Save the program.


Oracle Applications - Vision
File Edit View Folder Tools Window Help
Concurrent Programs

Program: SI SPML Processing ☑ Enabled
Short Name: SISPML
Application: Application Object Library
Description: HP SI SPML Processing for Application Reverse Agent

Executable
Name: SISPML
Method: SQL*Plus
Options:
Priority:

Request
Type:
Incrementor:
MLS Function:
☑ Use In SRS
□ Run Alone
□ Enable Trace
□ Allow Disabled Values
☑ Restart on System Failure
☑ NLS Compliant

Output
Format: Text
☑ Save (C)
☑ Print
Columns:
Rows:
Style:
□ Style Required
Printer:

Record: 1/1 <OSC>

18 Select the **Security** → **Responsibility** → **Request** form.

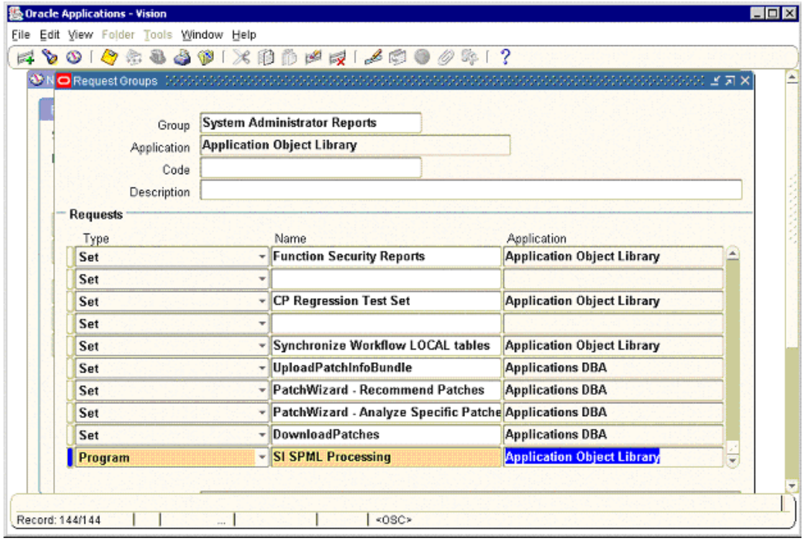
19 Find and choose the System Administrator Reports for the group.

20 At the end of the requests rows, add a new row by pressing TAB after the last field of last record.

21 Select **Program** as the type.

22 Enter the concurrent program defined for SPML processing above.

23 Save the form.


Oracle Applications - Vision
File Edit View Folder Tools Window Help
Request Groups

Group: System Administrator Reports
Application: Application Object Library
Code:
Description:

Requests

Type	Name	Application
Set	Function Security Reports	Application Object Library
Set		
Set	CP Regression Test Set	Application Object Library
Set		
Set	Synchronize Workflow LOCAL tables	Application Object Library
Set	UploadPatchInfoBundle	Applications DBA
Set	PatchWizard - Recommend Patches	Applications DBA
Set	PatchWizard - Analyze Specific Patche	Applications DBA
Set	DownloadPatches	Applications DBA
Program	SI SPML Processing	Application Object Library

Record: 144/144 <OSC>

Now the SYSADMIN user can schedule the SPML process as concurrent request.

24 Select the **Requests** → **Run** form.

25 Select **Single Request**.

26 Choose the SPML processing concurrent program defined in above for the name.

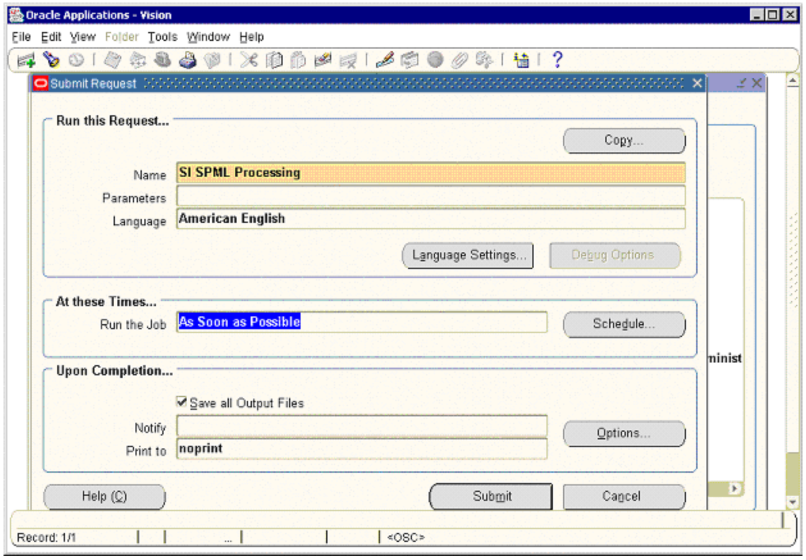
27 For troubleshooting purposes, select **As Soon As Possible** for on-demand processing. For production mode, schedule for periodic execution.

28 Click **Schedule**.

29 Select **Periodically** in the Run the Job section.

30 Select the job execution interval.

31 Save the request.



Submit Request

Run this Request...

Name: SI SPML Processing

Parameters:

Language: American English

At these Times...

Run the Job: As Soon as Possible

Upon Completion...

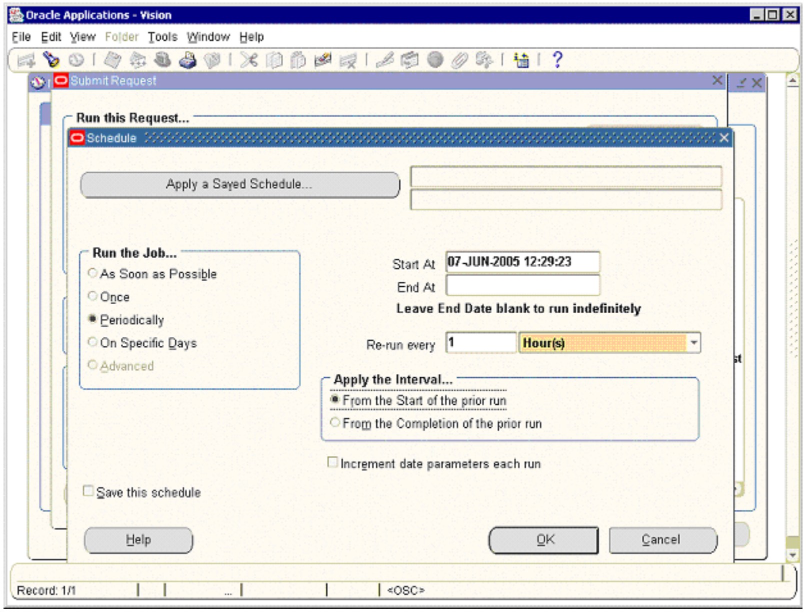
☑ Save all Output Files

Notify:

Print to: noprint

Record: 1/1 <OSC>

32 Submit the request.


Oracle Applications - Vision
File Edit View Folder Tools Window Help
Submit Request
Run this Request...
Apply a Saved Schedule...
Help
OK
Cancel

Schedule

Run the Job...

○ As Soon as Possible
○ Once
● Periodically
○ On Specific Days
○ Advanced

Start At: 07-JUN-2005 12:29:23

End At:

Leave End Date blank to run indefinitely

Re-run every: 1 Hour(s)

Apply the Interval...

● From the Start of the prior run
○ From the Completion of the prior run

□ Increment date parameters each run

□ Save this schedule

Record: 1/1 <OSC>

Define a Job Scheduler

You can also define an OS job scheduler to process SPML messages. This is done outside of the Oracle Application, using the OS features. Because the script is run in the database, any job scheduler with access to the Oracle Database client can execute the job.

For example, you could use the Oracle Enterprise manager management console. To process SPML messages, run the `SPMLProcess.sql` script with `SIConfig.sql` as the parameter. If the job scheduler is external to the database and uses a client-based SQL*Plus tool, the `SPMLProcess.sql` and `SIConfig.sql` files need to be accessible from the SQL*Plus client.

Custom Attributes for Employee Modifications

To provision additional attributes for employees, several modifications need to be made to the agent:

1. The mapping file must to be edited to include the additional attributes.
2. The event alert must be defined to monitor the attribute changes.
3. The `SPMLProcess.sql` script must be changed to incorporate the logic for additional attributes.

The following attributes are added in the example procedure below:

- EMP_REVIEW_DATE
- EMP_REVIEW_RATING

The following example describes the steps for adding employee attributes to track performance review dates and ratings:

1. Add the attributes to the mapping file (`ORAERPEMP-11-5-9.xml` for version 11.5.9 or `ORAERPEMP-11-5-10.xml` for version 11.5.10).

   In the mapping file, the <attributeDefinitionReference> block and <attributeDefinition> block must be defined. Here are examples for the attributes listed above:

```
<attributeDefinitionReference name="ATTR_EMP_Review_Date"
required="false" concero:tafield="EMP_Review_Date"
concero:resfield="[x_emp_review_date][EMP_REVIEW_DATE][][DATE]"
concero:init="true" />

<attributeDefinitionReference name="ATTR_EMP_Review_Rating" required="false"
concero:tafield="EMP_Review_Rating"
concero:resfield="[x_emp_review_rating][EMP_REVIEW_RATING][][NUMBER]"
concero:init="true" />

<attributeDefinition name="ATTR_EMP_Review_Date" description="Employee Review
Date" type="xsd:date">
   <properties>
      <attr name="minLength">
         <value>1</value>
      </attr>
      <attr name="maxLength">
         <value>64</value>
      </attr>
   </properties>
</attributeDefinition>
```

```
<attributeDefinition name="ATTR_EMP_Review_Rating" description="Employee Review
Rating" type="xsd:string">
   <properties>
      <attr name="minLength">
         <value>1</value>
      </attr>
      <attr name="maxLength">
         <value>64</value>
      </attr>
   </properties>
</attributeDefinition>
```

2 Modify `oracle11ierp.xsl` to add the attribute information for reverse provisioning. Note the attribute name is lowercase for the first line.

```
<xsl:when test="$ATTRNAME = 'emp_review_date'">
   <xsl:call-template name="AttributeBuilder">
      <xsl:with-param name="DSMLELEMENT" select="$DSMLELEMENT"/>
      <xsl:with-param name="ATTRNAME" select="'EMP_Review_Date'"
       />
      <xsl:with-param name="ATTRVALUE" select="$ATTRVALUE" />
      <xsl:with-param name="MODIFYFLAG" select="$MODIFYFLAG" />
   </xsl:call-template>
</xsl:when>

<xsl:when test="$ATTRNAME = 'emp_review_rating'">
   <xsl:call-template name="AttributeBuilder">
      <xsl:with-param name="DSMLELEMENT" select="$DSMLELEMENT"/>
      <xsl:with-param name="ATTRNAME"
       select="'EMP_Review_Rating'" />
      <xsl:with-param name="ATTRVALUE" select="$ATTRVALUE" />
      <xsl:with-param name="MODIFYFLAG" select="$MODIFYFLAG" />
   </xsl:call-template>
</xsl:when>
```

3 Modify the Select Identity resource and Service with the new mapping file and map the new attributes.

4 Modify the `SIConfig.sql` file to add the new attribute name and value container.

```
/* Attribute name for employee review date */
SIEMPREVIEWDATE VARCHAR2(32) := 'EMP_Review_Date';

/* Attribute name for employee review rating */
SIEMPREVIEWRATING VARCHAR2(32) := 'EMP_Review_Rating';

/* Value container for review date. Format is YYYY-MM-DD.
 * TO_CHAR conversion of date value is nessary for SPML
 * transmission
 */
EMPREVIEWDATE VARCHAR2(32);

/* Value container for review rating. */
EMPREVIEWRATING VARCHAR2(32);
```

5 Define a new event alert to monitor the performance review changes. The table used for employee performance review is PER_PERFORMANCE_REVIEWS. Create event alert on PER_PERFORMANCE_REVIEWS. The event alert should select the PERSON_ID. Refer to Event Alert Definitions on page 21 for details on creating event alert for employees.

Application Name: Human Resources
Table Name: PER_PERIODS_OF_SERVICE
Event to Monitor: After Update After Insert
Select SQL Script:
    SELECT PERSON_ID

```
INTO &PERSON_ID
FROM PER_PERFORMANCE_REVIEWS
WHERE ROWID = :ROWID;
```

Action SQL Script:

```
INSERT INTO sioraerp.user_requests (request_num, ID, user_type,
 action_type )
values (sioraerp.user_request_seq.nextval, &PERSON_ID, 'E', 'M' );
COMMIT;
```

6 Modify the `SPMLProcess.sql` script to include the logic to retrieve the attribute values. Locate the following comment in the file:

```
/* Start Addtional Attributes Here */
```

Insert the SQL code to retrieve the custom attribute values for the employee. The user ID is contained in provision_req_rec.USER_ID cursor variable.

Define the MAX_REVIEW_DATE variable in the declaration section in the top:

```
MAX_REVIEW_DATE   Date;
```

```
/* Get the latest review date for the employee */
SELECT MAX(REVIEW_DATE)
INTO MAX_REVIEW_DATE
FROM _PERFORMANCE_REVIEWS
WHERE PERSON_ID = provision_req_rec.USER_ID;

SELECT TO_CHAR(PERFORMANCE_RATING)
INTO EMPREVIEWRATING
FROM _PERFORMANCE_REVIEWS
WHERE PERSON_ID= provision_req_rec.USER_ID
AND REVIEW_DATE = MAX_REVIEW_DATE;

/* Convert the date to string */
SELECT TO_CHAR(MAX_REVIEW_DATE,'YYYY-MM-DD')
INTO EMPREVIEWDATE
FROM DUAL;

/* Assign the values to SPML attributes */
attab(SIEMPREVIEWDATE) := EMPREVIEWDATE;
attab(SIEMPREVIEWRATING) := EMPREVIEWRATING;
```

7 Test the new attributes. When performance review changes occur, the new event alert should fire. Check the Alert Request View window to make sure that the event alert executed. Then, check the contents of sioraerp.user_requests table by issuing the following SQL:

```
Select * from sioraerp.user_requests;
```

The user request table should contain the person ID of the employee review performed and 'M' for action type flag.

Run `SPMLProcess.sql` in SQL command window. It is recommended to run the script in the database server command window. Check the SPML message sent back to Select Identity. The SPML message can be found in the sioraerp.provision_requests spml_message column.

Secure Communication for Reverse Provisioning

The agent can communicate with the Select Identity Web Service over a secure communication channel (SSL) using the HTTPS protocol. To enable the SSL communication, perform the following steps:

1. Obtain a trusted certificate for the web server. Trusted certificates can be obtained from Certificate Authority such as Verisign or can be generated internally using the OpenSSL toolkit or Microsoft Certificate Authority (CA) Server. The following provides an example of how to generate a certificate:

   a. Set the environment for WebLogic by running `setenv.sh` (or `setenv.cmd`) on the WebLogic server domain.

   b. Run `keytool` to generate a keystore on the WebLogic server. Here is an example command:

   ```
   keytool -genkey -v -keyalg rsa -keysize 1024
   -keypass abcd1234 -keystore helix.jks
   -storepass abcd1234 -storetype jks
   ```

   c. Run `keytool` to generate the certificate request:

   ```
   keytool -certreq -v -keyalg rsa -file sidemo.csr -keypass abcd1234
   -keystore helix.jks
   -storepass abcd1234 -storetype jks
   ```

   d. Generate the certificate using the Microsoft CA server from the request generated.

   e. e.Export the certificate generated from MS CA server. Export the certificate in p7b format including the entire certificate chain.

   f. Export the root CA certificate in base 64 format.

   g. Export the server certificate.

   h. Copy the certificates to the WebLogic server.

   i. Run `keytool` to import the certificate:

   ```
   keytool -import -v -trustcacerts -alias mykey
   -keyalg rsa -file helix.p7b  -keypass abcd1234 -keystore helix.jks
   -storepass abcd1234 -storetype jks
   ```

   j. Use the keystore to configure WebLogic SSL.

   k. Import the root CA certificate and server certificate in base 64 format to Oracle Wallet.

   l. Copy the Oracle Wallet file (`ewallet.p12`) to the Oracle Application database server directory (`/app/owm`).

   m. Configure the URL for secure WebLogic web service in the `SIConfig.sql` file and run `updateowminfo.sql`.

2. Configure WebLogic to accept SSL communication with the certificate. A certificate keystore needs to be generated from the trusted certificate for use with WebLogic. Keytools can be used to generate the keystore.

3. Specify the URL in `SIConfig.sql` (such as https://weblogichost:7002/lmz/webservice).

4. To enable the Oracle 11i agent to use SSL communication, Oracle Wallet must be created. A wallet is a container that is used to store authentication and signing credentials, including private keys, certificates, and trusted certificates needed by SSL. In an Oracle

environment, every entity that communicates over SSL must have a wallet containing an X.509 version 3 certificate, private key, and list of trusted certificates. Oracle Wallet manager is installed as part of Oracle 9i client tools. Launch Oracle Wallet Manager by selecting **Start** → **Programs** → **Oracle 92** → **Intergrated Management Tools** → **Wallet Manager**.

5 Create a new wallet by clicking **New**.

6 Enter the wallet password. This is password opens the wallet and is not associated with other passwords. This password must then be supplied to the agent to enable the agent to open the wallet.

7 When prompted, click **No** to create a certificate request.

8 Select **Operations** → **Import Trusted Certificate**.

9 Import the trusted certificate and the root certificate of the certificate authority. It is important to import the full certificate chain.

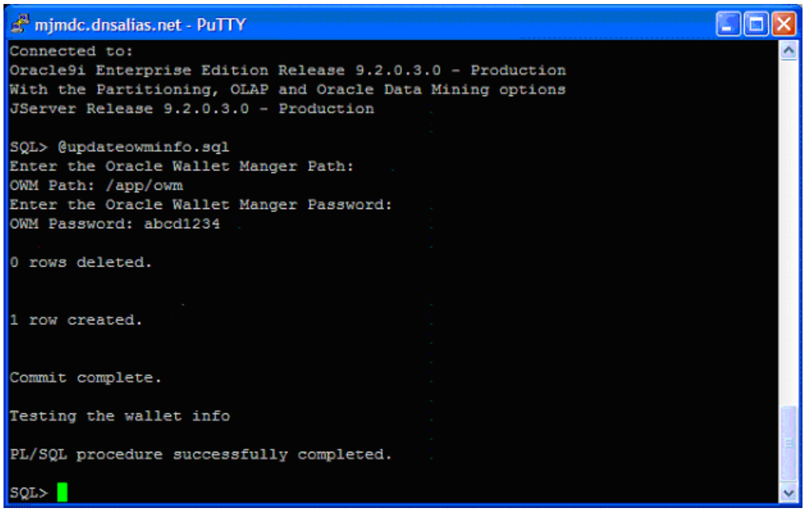
10 Save the wallet file in the wallet directory (such as c:\owm). The wallet file is called `ewallet.p12`.

11 Create a wallet directory on the Oracle Application database server. The wallet command is issued by the database and the wallet directory must be accessible by the database server, not the application forms server. Only one wallet file can reside per wallet directory. Thus, if there is existing wallet directory, create a new one.

12 Copy the `ewallet.p12` file to the Oracle Applications database server wallet directory.

13 Go to the directory where the `SIConfig.sql` file resides.

Run `updateowminfo.sql` as the sioraerp or apps user in SQL Plus, which will prompt for the wallet directory and the wallet password. The script then tests wallet certificate functionality by calling HTTPS using the URL configured in `SIConfig.sql`.



```
Connected to:
Oracle9i Enterprise Edition Release 9.2.0.3.0 - Production
With the Partitioning, OLAP and Oracle Data Mining options
JServer Release 9.2.0.3.0 - Production

SQL> @updateowminfo.sql
Enter the Oracle Wallet Manger Path:
OWM Path: /app/owm
Enter the Oracle Wallet Manger Password:
OWM Password: abcd1234

0 rows deleted.


1 row created.


Commit complete.

Testing the wallet info

PL/SQL procedure successfully completed.

SQL>
```

f the wallet certificate is valid, the PL/SQL procedure will complete without any error. Address any errors that occur, such as an incorrect wallet path, incorrect password, or invalid certificate chain. (Invalid certificate chain usually means incorrect certificates or incomplete certificate chain.) Both the server certificate and CA root certificate must be imported into the wallet.

14 Modify `SIConfig.sql` to specify HTTPS as the protocol for the URL for the Web Service.

Configuring a Secure JDBC Connection Pool

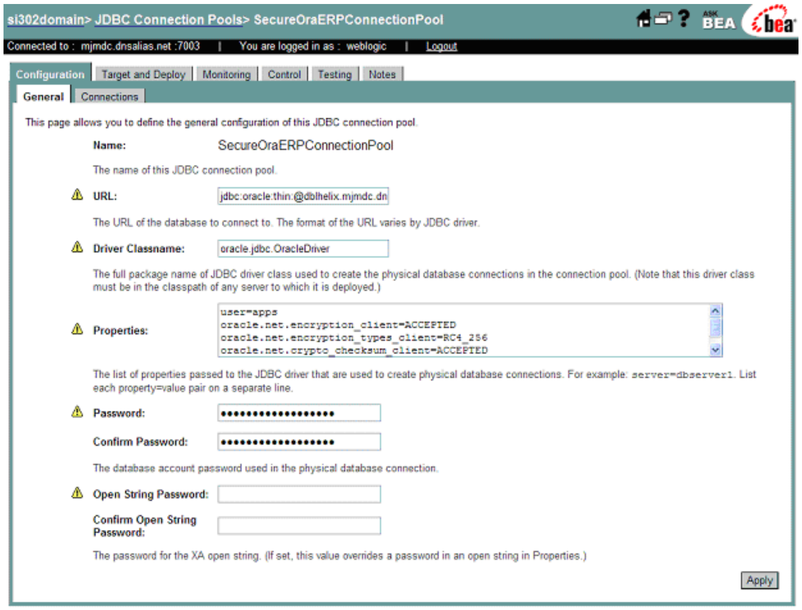
To enable secure communication between Select Identity and the Oracle E-Business Suite, you must configure connection properties for the Oracle JDBC driver. You configure these properties in the WebLogic JDBC connection pool, in the Properties attribute. This attribute is available on the **JDBC Connection Pool** → **Configuration** → **General** tab on the Administration Console. You can set the following properties to enable encryption for the connection pool:

oracle.net.encryption_client=ACCEPTED

oracle.net.encryption_types_client=RC4_256

oracle.net.crypto_checksum_client=ACCEPTED

Here is a snapshot of an example configuration:



si302domain> JDBC Connection Pools> SecureOraERPConnectionPool
ASK BEA
bea
Connected to : mjmdc.dnsalias.net :7003 | You are logged in as : weblogic | Logout
Configuration
Target and Deploy
Monitoring
Control
Testing
Notes
General
Connections
Apply

This page allows you to define the general configuration of this JDBC connection pool.

Name: SecureOraERPConnectionPool

The name of this JDBC connection pool.

⚠ **URL:** jdbc:oracle:thin:@dblhelix.mjmdc.dn

The URL of the database to connect to. The format of the URL varies by JDBC driver.

⚠ **Driver Classname:** oracle.jdbc.OracleDriver

The full package name of JDBC driver class used to create the physical database connections in the connection pool. (Note that this driver class must be in the classpath of any server to which it is deployed.)

⚠ **Properties:**

```
user=apps
oracle.net.encryption_client=ACCEPTED
oracle.net.encryption_types_client=RC4_256
oracle.net.crypto_checksum_client=ACCEPTED
```

The list of properties passed to the JDBC driver that are used to create physical database connections. For example: `server=dbserver1`. List each property=value pair on a separate line.

⚠ **Password:** ●●●●●●●●●●●●●●●●●●

Confirm Password: ●●●●●●●●●●●●●●●●●●

The database account password used in the physical database connection.

⚠ **Open String Password:**

Confirm Open String Password:

The password for the XA open string. (If set, this value overrides a password in an open string in Properties.)

Generating Files for the User Import Function

The User Import function provided by Select Identity enables you to add a group of users that exist on the resource to Select Identity. The list of users and their associated attributes are specified in an SPML file, which is processed by the Select Identity server to add the users. If you want the Oracle 11i connector's `SPMLProcess.sql` script to generate the User Import SPML file instead of sending individual requests to the Select Identity Web Service, complete the following steps:

1 Edit the following parameters in `SIConfig.sql` file:

— SIAUTODISCOVERY — Set the value of this parameter to **1**, which instructs the `SPMLProcess.sql` procedure to create the SPML file rather than sending add requests to the Select Identity Web Service. Here is an example:

```
SIAUTODISCOVERY NUMBER := 1; /* 1 to turn on autodiscovery message.  0 for
reconciliation web service */
```

— SIAUTOXMLDIR — Set the value of this parameter to the location where the SPML file will reside. The directory must exist on the database server (not the application server). Specify the file name in the following format:

```
auto-RESOURCE_NAME-File_Sequence.xml
```

where *RESOURCE_NAME* is the foundation and employee resource specified by the SIFNDRESOURCE and SIEMPRESOURCE parameters. Here is an example:

```
SIAUTOXMLDIR VARCHAR2(256) := '/tmp'; /* Directory for User Import XML File */
```

— MAXAUTOXMLUSER — Set the value of this parameter to the maximum number of users that can be included in each User Import SPML file. When the user count exceeds the maximum number, a new SPML file is created. Here is an example:

```
MAXAUTOXMLUSER  NUMBER := 500;  /* Maximum number of users per autoxml file */
```

2 Populate the `sioraerp.user_requests` table with the foundation and employee user information. Make sure that the MAXID parameter in `perfemp.sql` and `perffnd.sql` is larger than the user or employee count. Run `perfemp.sql` to populate the `sioraerp.user_request` table with employee information. Run `perffnd.sql` to populate the `sioraerp.user_request` table with application user information.

3 Run `SPMLProcess.sql`, which resides in the Select Identity installation directory on the Oracle server; run this file as the apps user. The script takes `SIConfig.sql` as an input parameter. Make sure the environment variable is set correctly. Here is an example of the commands you can issue to run this file:

```
cd SI_install_dir

sqlplus apps/apps

@perfemp.sql

@perffnd.sql

@SPMLProces 'SIConfig.sql'
```

The User Import SPML file is saved in the specified directory on the database server. When the `SPMLProcess.sql` file runs, any SPML files in that directory are overwritten; move them to another directory to archive them before running `SPMLProcess.sql`. Also, the `sioraerp.user_requests` table is cleaned out after `SPMLProcess.sql` runs. To recreate the SPML files, repeat step 2 and step 3.

4 After the User Import SPML files are generated, set the SIAUTODISCOVERY parameter to **0** to disable generation of SPML files and enable Web Service reconciliation.

5 Configuring the Connector with Select Identity

This chapter describes the procedure to configure the Oracle 11i connector with Select Identity and the connector specific parameters that you must provide while configuring the connector with Select Identity.

Configuration Procedure

After you deploy the connector RAR on application server, you must configure the connector with Select Identity. Perform the following steps to configure the Oracle 11i connector with Select Identity.

1. Add a New Connector
2. Add a New Resource
3. Map Attributes

Add a New Connector

Add a new connector in Select Identity by using the user interface. While adding the connector, do the following:

- In the Connector Name text box, specify a name for the connector.
- In the Pool Name text box, enter **`eis/ORAERP`**.
- Select No for the Mapper Available section.

Refer to the *HP Select Identity Connector Deployment Guide* for detailed information on adding a new connector in Select Identity.

Add a New Resource

Add a new resource in Select Identity that uses the newly added connector. Refer to the *HP Select Identity Connector Deployment Guide* for detailed instructions on adding a resource in Select Identity.

Refer to the following table while entering the parameters in the Basic Information and the Access Information pages:

Table 5 Resource Configuration Parameters

Field Name	Sample Values	Description	Comment
Resource Name	oracle11i	Name given to the resource. If you enabled reverse synchronization, this must be the same as the value provided for the urn:trulogica:concero:2.0#resourceId attribute on the agent console.	
Connector Name	Oracle 11i	The newly deployed connector.	Known as Resource Type on Select Identity 3.3.1.
Authoritative Source	No	Whether this resource is a system that is considered to be the authoritative source for user data in your environment. Specify **No** if the connector is not enabled for reverse synchronization. Specify **Yes** if you want to add users through reverse synchronization. If the resource is not authoritative, the resource can only modify user entitlements during reverse synchronization.	
Associate to Group	Selected	Whether the system uses the concept of groups. For the Oracle 11i connector, select this option.	Applicable only on Select Identity 3.3.1.

Table 5 Resource Configuration Parameters (cont'd)

Field Name	Sample Values	Description	Comment
DataSource Name	jdbc/oracle11i	The JDBC data source for the Oracle 11i database, if you are using a data source to connect to the database. If you are using WebSphere 5.1.1 as the application server, the JDBC data source for the Oracle database will not work; therefore, you must use the JDBC URL and driver to connect to the database.	
URL	jdbc:oracle:thin:@kod:1522:VIS	The JDBC URL to the Oracle database, if you are using a JDBC driver to connector to the database.	
DriverClass Name	oracle.jdbc.Oracle Driver	The class name of the Oracle JDBC driver, if you are using a JDBC driver to connector to the database. If you are using the WebLogic application server, use Oracle's Thin driver, which is listed in the drop down list of WebLogic. Both the data source and JDBC URL and driver connections can be used with this driver. If you are using WebSphere 5.1.1, download the latest Oracle Thin driver (`ojdbc14.jar`), version 10.1.0.4.0 or later.	

Table 5 Resource Configuration Parameters (cont'd)

Field Name	Sample Values	Description	Comment
UserID	apps	The Oracle schema owner.	
Password	apps	The password of the schema owner.	
User Table Name	FND_USER	The Oracle database table where foundation user information is stored. Do not change this value.	
Group Table Name	FND_ RESPONSIBILITY	The name of the Oracle table where responsibility information is stored. Do not change this value.	

Table 5 Resource Configuration Parameters (cont'd)

Field Name	Sample Values	Description	Comment
Link Table Name	FND_USER_RESP_ GROUPS	The name of the Oracle table that maps users to responsibilities. Do not change this value.	
Mapping File	ORAERP-11-5-9.xml	The name of the mapping file that maps Select Identity fields to Oracle fields. If you are provisioning in an Oracle E-Business Suite 11i server, specify **ORAERP-11-5-9.xml** or **ORAERP-11-5-10.xml**, depending on the version of Oracle. If you are provisioning in an Oracle Human Resources system, specify **ORAERPEMP-11-5-9.xml** or **ORAERPEMP-11-5-10.xml**, depending on the version of Oracle.	
Is Employee	Yes	For an employee, set this parameter to **Yes**. For other (normal) users, set this parameter to **No**.	

*Instead of creating an authoritative resource, you can create authoritative attributes (in the next step) for the attributes that will be synchronized. Entitlements are authoritative by default in a non-authoritative resource but other attributes are not.

Map Attributes

After successfully adding a resource for the Oracle 11i connector, you must map the resource attributes to Select Identity attributes. Refer to the *HP Select Identity Connector Deployment Guide* for information on mapping and creating attributes. While mapping attributes, refer to the following table for resource specific mapping information.

ORAERP-11-5-*x*.xml Mapping Information

The following attributes are listed in the `ORAERP-11-5-x.xml` mapping file. You can add, modify, and delete attributes once you are familiar with the contents of this file.

The Select Identity resource attributes are editable. They reflect the identity information as seen in Select Identity. The physical resource attributes are literal attributes of user accounts on Oracle. These attributes cannot be changed. The user name attribute is not case sensitive in Oracle, though it is necessary to make the user name attribute all uppercase to be compatible with Oracle and Select Identity

Table 6 Oracle 11i Mapping Information

Select Identity Attribute				**Oracle 11i Resource Attribute**
Name	**Type**	**Min Length**	**Max Length**	**Name**
UserName*	String	1	100	UserName
Description	String	1	240	Desc
customerID	String	1	15	CustomerId
supplierID	String	1	15	SupplierID
EmployeeID	String	1	15	EmployeeID
Email*	String	1	240	Email
Owner	String	1	64	Owner
Fax	String	1	80	Fax
PasswordLifespan Days	String	1	15	Days
Password Accesses	String	1	15	Accesses
StartDate	Date	1	64	StartDate
EndDate	Date	1	64	EndDate
Password*	String	1	64	Password
ICX_HR_PERSON_ID**	String	1	15	ICX_HR_ PERSON_ID**
TO_PERSON_ ID**	String	1	15	TO_PERSON_ID**

* Select Identity default attribute
** Oracle Application securing attribute

In the table above, some Oracle 11i attributes are securing attributes. These attributes are used by Oracle Self-Service Web Applications to allow visibility to specific users or responsibilities for rows (records) of data based on the specific data (attribute values) contained in the row. Because securing attributes are dynamic and differ from site to site, the specific securing attributes used for each site need to be contained in the XML mapping file.

Each Select Identity attribute name that maps to an Oracle Application securing attribute must match the Oracle Applications's ATTRIBUTE_CODE for reverse synchronization purposes (such as ICX_HR_PERSON_ID) . This ATTRIBUTE_CODE is also placed in the Column field of the attribute in `ORAERP-11-5-x.xml` file. Here is an example:

```
<attributeDefinitionReference name="ATTR_ICX_HR_PERSON_ID" required="false"
concero:tafield="ICX_HR_PERSON_ID"
concero:resfield="[p_attribute_code][ICX_HR_PERSON_ID]
[Oracle Self-Service Web Applications][NUMBER]" concero:init="true" />

<attributeDefinitionReference name="ATTR_TO_PERSON_ID" required="false"
concero:tafield="TO_PERSON_ID"
concero:resfield="[p_attribute_code][TO_PERSON_ID]
[Oracle Self-Service Web Applications][NUMBER]"
concero:init="true" />
```

The resfield attributes consists of four values:
[parameter name][COLUMN_NAME][Application Name][VALUE_TYPE]

where:

- parameter name — The name of the parameter used by Oracle's PL/SQL package. Standard attributes that map to Oracle Application's Foundation user attributes must specify the parameter name used by FND_USER packages. For securing attributes, parameter names will always be p_attribute_code.
- COLUMN_NAME — The name of the column that each attribute maps to Oracle's table. Standard attributes that map to Oracle Application's Foundation user attributes must specify the column name in FND_USER table. For securing attributes, specify ATTRIBUTE_CODE for each attribute.
- Application Name — Leave blank for standard attributes. For securing attributes, specify the Oracle Application Name for each securing attribute. The Application Name may differ for different releases of Oracle Application. The Application Name can be identified in the Oracle Application user definition window's securing attribute section. If the wrong Application Name is specified, actions against securing attributes will fail. For example, on Oracle App 11.5.9, the application name is Oracle Self-Service Web Applications. On Oracle App 11.5.10, the application name is Self-Service Web Applications.
- VALUE_TYPE — Specify one of the following: VARCHAR, DATE, or NUMBER.

ORAERPEMP-11-5-*x*.xml Mapping Information

The following attributes are used for provisioning employees in the Oracle Human Resources system. These are listed in the `ORAERPEMP-11-5-x.xml` mapping file. You can add, modify, or delete attributes once you are familiar with the contents of this file.

The Select Identity resource attributes are editable. They reflect the identity information as seen in Select Identity. The physical resource attributes are literal attributes of user accounts in the Oracle Human Resources system. These attributes cannot be changed. The user name attribute is not case sensitive in Oracle, though it is necessary to make the user name attribute all uppercase to be compatible with Oracle and Select Identity.

Table 6A Oracle 11i Mapping Information

Select Identity Attribute					Oracle 11i Resource Attribute
Name	**Description**	**Type**	**Min Length**	**Max Length**	**Name**
UserName*	User name derived from first three characters of last name plus the person ID	String	1	100	UserName
Description	Full name	String	1	240	Desc
customerID		String	1	15	CustomerId
supplierID		String	1	15	SupplierID
EmployeeID	Person ID	String	1	15	*EmployeeID*
Email*		String	1	240	Email
Owner		String	1	64	Owner
Fax		String	1	80	Fax
PasswordLifespan Days		String	1	15	Days
Password Accesses		String	1	15	Accesses
StartDate		Date	1	64	*StartDate*
EndDate	Termination date	Date	1	64	*EndDate*
Password*	Default set in XSL file	String	1	64	Password
ICX_HR_PERSON_ID**	Person ID	String	1	15	ICX_HR_ PERSON_ID **
TO_PERSON_ID**	Person ID	String	1	15	TO_PERSON _ID**
CompanyName	GRE (business group name)	String	1	240	*Company Name*
OrganizationName	Organization (department) name	String	1	240	*Organization Name*
FirstName		String	1	150	*FirstName*

Table 6A Oracle 11i Mapping Information (cont'd)

Select Identity Attribute					**Oracle 11i Resource Attribute**
LastName		String	1	150	*LastName*
EmployeeNumber	Employee number for organization	String	1	64	*Employee Number*
DOB	Date of birth in this format: YYYY-MM-DD	Date	1	64	*DOB*
Addr1	Home address, line 1	String	1	240	*Addr1*
Addr2	Home address, line 2	String	1	240	*Addr2*
City	Home address city	String	1	30	*City*
State	Home address state	String	1	120	*State*
Zip	Home address postal code	String	1	30	*Zip*
Country	Home address country	String	1	60	*Country*
Job	Job name	String	1	700	*Job*
Location	Job location	String	1	240	*Location*
Grade	Pay grade	String	1	240	*Grade*
Position	Job Position	String	1	240	*Position*
ManagerFlag	Y for manager, N for employee	String	1	10	*ManagerFlag*
*WorkPhone***		String	1	64	*WorkPhone*
HomePhone		String	1	64	*HomePhone*

* Select Identity default attribute
** Oracle Application securing attribute (see page 77 for more information)

Italicized attributes are monitored for changes in the Oracle Human Resources system by the Oracle Alert system. Changes to the monitored employee attributes are communicated to Select Identity by the agent. Synchronizing additional attributes can be accomplished by adding the attribute information in the `ORAERPEMP-11-5-x.xml` mapping file and the `oracle11ierp.xsl` reverse mapping file after creating the Oracle Alert to monitor the desired employee attributes.

After mapping the attributes, you can use the connector to create a service, or you can associate the connector with an existing service. Refer to the *Service Studio* chapter of the *HP Select Identity Administration Online Help* for information on Select Identity services.

6 Uninstalling the Connector

If you want to uninstall a connector from SI, perform the following steps:

- Remove all resource dependencies.
- Delete the connector from Select Identity.
- Delete the connector from application server.
- Uninstall the agent.

See *HP Select Identity Connector Deployment Guide* for more information on deleting the connector from application server and Select Identity.

Uninstalling the Agent

To uninstall the Oracle 11i agent, perform following steps:

1. Deactivate all alerts defined during the agent installation. If alerts were fired, the alerts cannot be deleted.
2. Deactivate the Periodic Alerts for SPML processing if defined.
3. Deactivate the Concurrent Request processing for SPML processing if defined.
4. Log on to SQLPLUS as system account and issue following statements:

```
drop user sioraerp cascade;

drop package apps.provision_service;
```

A Troubleshooting

To troubleshoot agent issues, you can examine the user_requests and provision_requests tables. Also, the `SPMLProcess.sql` file can be run manually to produce debugging output.

- User_requests table

  The siroraerp.user_requests table contains the output from each event alert. In the Alert History window in Oracle application, the result of the alert action can be examined. If the alert completed without any error, the contents of the user_requests table can be examined to ensure that proper records are inserted into the table. During this process, the periodic alert to run `SPMLProcess.sql` may need to be stopped because the script will remove all the entries from the user_requests table after updating the provision_requests table.

- Provision_requests table

  The provision_requests table contains three columns:

  - spml_message — contains the outgoing spml message.
  - response — contains response from Select Identity.
  - error_msg — contains any error message if the SPML message results in an error.

  Also, the req_status field is set to -1 for error. The retry column contains the number of retries for the specific SPML message. To reprocess the user action, the req_status field can be set to 0.

The user_requests and termination_request tables are temporary and do not store historical information. The provision_requests table, however, does retain all SPML activities. The contents of provision_requests table can be used as an audit trail or for reporting purposes. The provision_requests table can also be purged. To purge the table, run following sql statement as the sioraerp user.

```
DELETE FROM PROVISION_REQUEST

WHERE REQ_STATUS IN ( -1, 2 );

      COMMIT;
```

Finally, one of the easiest ways to debug the agent is to run the `SPMLProcess.sql` script manually in SQL Plus. In the Select Identity installation directory on the Oracle server, run `SPMLProcess.sql` script as the apps user. The script takes `SIConfig.sql` as an input parameter.