Oracle® Identity Governance

Configuring the Oracle E-Business Suite User Management Application

12c (12.2.1.3.0)
E90987-04
October 2020

Oracle Identity Governance Configuring the Oracle E-Business Suite User Management Application, 12c (12.2.1.3.0)

E90987-04

Primary Author: Gowri.G.R

Contributors: Samriti Gupta

Contents

List of Figures

List of Tables

Preface

This guide describes the connector that is used to onboard Oracle E-Business Suite applications to Oracle Identity Governance.

Audience

This guide is intended for resource administrators and target system integration teams.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For information about installing and using Oracle Identity Governance 12.2.1.3.0, visit the following Oracle Help Center page:

`https://docs.oracle.com/en/middleware/idm/identity-governance/12.2.1.3/index.html`

For information about installing and using Oracle Identity Manager 11.1.2.3, visit the following Oracle Help Center page:

`http://docs.oracle.com/cd/E52734_01/index.html`

For information about Oracle Identity Governance Connectors 12.2.1.3.0 documentation, visit the following Oracle Help Center page:

`https://docs.oracle.com/en/middleware/idm/identity-governance-connectors/12.2.1.3/index.html`

For information about Oracle Identity Manager Connectors 11.1.1 documentation, visit the following Oracle Help Center page:

`http://docs.oracle.com/cd/E22999_01/index.htm`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New In This Guide?

These are the updates made to the software and documentation for release 12.2.1.3.0 of Configuring the Oracle E-Business Suite User Management Application.

The updates discussed in this chapter are divided into the following categories:

- Software Updates

  This section describes updates made to the connector software.

- Documentation-Specific Updates

  These include major changes made to the connector documentation. These changes are not related to software updates.

Software Updates

These are the updates made to the connector software.

Software Updates in Release 12.2.1.3.0

The following is the software update in release 12.2.1.3.0:

Support for Onboarding Applications Using the Connector

From this release onward, the connector bundle includes application onboarding templates required for performing connector operations on the Oracle E-Business Suite User Management target system. This helps in quicker onboarding of the applications for this target system into Oracle Identity Governance by using an intuitive UI.

Documentation-Specific Updates

These are the updates made to the connector documentation.

Documentation-Specific Updates in Release 12.2.1.3.0

The following documentation-specific update has been made in revision "04" of this guide:

The "Target system" row of Table 1-1 has been updated to include a note about applying an Oracle Database patch. In addition, Oracle Database 19c has been added as one of the supported versions for running the target system.

The following documentation-specific update has been made in revision "03" of this guide:

The "Oracle Identity Governance or Oracle Identity Manager" row of Table 1-1 has been updated to remove Oracle Identity Manager versions prior to 11g Release 2 PS3 (11.1.2.3.0).

The following documentation-specific update has been made in revision "02" of this guide:

The "Oracle Identity Governance or Oracle Identity Manager" row of Table 1-1 has been updated to include support for Oracle Identity Governance release 12*c* PS4 (12.2.1.4.0).

This is the first release of the connector. Therefore, there are no documentation-specific updates in this release.

1

About the Oracle E-Business Suite User Management Connector

The Oracle E-Business Suite User Management connector integrates Oracle Identity Governance with Oracle E-Business Suite.

The following topics provide a high-level overview of the connector:

- Introduction to the Connector
- Certified Components
- Usage Recommendation
- Certified Languages
- Supported Connector Operations
- Connector Architecture
- Supported Connector Features Matrix
- Connector Features

> **Note:**
>
> In this guide, Oracle E-Business Suite User Management connector is referred to as the **EBS User Management connector**.

1.1 Introduction to the Connector

Oracle Identity Governance is a centralized identity management solution that provides self service, compliance, provisioning and password management services for applications residing on-premises or on the Cloud. Oracle Identity Governance connectors are used to integrate Oracle identity Governance with the external identity-aware applications. The Oracle E-Business Suite User Management connector lets you onboard Oracle E-Business Suite applications in Oracle Identity Governance.

> **Note:**
>
> In this guide, the connector that is deployed using the **Applications** option on the **Manage** tab of Identity Self Service is referred to as an **AOB application**. The connector that is deployed using the **Manage Connector** option in Oracle Identity System Administration is referred to as a **CI-based connector** (Connector Installer-based connector).

From Oracle Identity Governance release 12.2.1.3.0 onward, connector deployment is handled using the application onboarding capability of Oracle Identity Self Service. This capability lets business users to onboard applications with minimum details and effort. The connector installation package includes a collection of predefined templates (XML files) that contain all the information required for provisioning and reconciling data from a given application or target system. These templates also include basic connectivity and configuration details specific to your target system. The connector uses information from these predefined templates allowing you to onboard your applications quickly and easily using only a single and simplified UI.

Application onboarding is the process of registering or associating an application with Oracle Identity Governance and making that application available for provisioning and reconciliation of user information.

An FND_USER record represents an Oracle E-Business User Management account. This record is the main component of the account data whose management is enabled by the connector. This connector can be used to manage either the FND_USER records or FND_USER records with TCA records. In other words, this connector is used to manage plain user accounts or user accounts with parties.

You can use the User Management connector to create Oracle E-Business Suite user accounts (FND_USER records) for OIG users and to grant user roles and responsibilities to these accounts. You can also reconcile newly created users and modified user accounts (FND_USER records) from the target system. These reconciled records are used to create and update Oracle E-Business User Management accounts assigned to OIG Users.

In addition to creating Oracle E-Business User Management accounts, you can use this connector to create Party or Vendors (Suppliers) in the target system. Party or vendors represent a Trading Community Architecture (TCA) record in the HZ_PARTIES table. Some applications such as iStore, iProcurement in the Oracle E-Business Suite require users to have a TCA record that is a representative or employee of parties and vendors in your organization.

The following are the types of TCA records that this connector supports:

- Parties
- Vendors or Suppliers

The object class used for the User Management connector with TCA party is `__ACCOUNT__`. Roles and responsibilities are handled as child data. You can use this connector to remove existing roles and responsibilities as well.

During user provisioning, if you enter the party or supplier information along with the EBS user information, the connector creates an E-Business user account first, creates the party or vendor next, and then establishes the link between the user record and TCA record. For target system users that are linked with party or Supplier records, the value in the PERSON_PARTY_ID column in the FND_USER table is the same as the value in the PARTY_ID column of the HZ_PARTIES table.

During a create or update user provisioning operation, you can link the target system user account with an existing HRMS employee record by providing Person ID.

1.2 Certified Components

These are the software components and their versions required for installing and using the connector.

> **Note:**
>
> If you are using Oracle Identity Manager release 11.1.x, then you can install and use the connector only in the CI-based mode. If you want to use the AOB application, then you must upgrade to Oracle Identity Governance release 12.2.1.3.0 or later.

Table 1-1 Certified Components

Component	Requirement for AOB Application	Required for CI-Based Connector
Oracle Identity Governance or Oracle Identity Manager	You can use one of the following releases of Oracle Identity Governance or Oracle Identity Manager: • Oracle Identity Governance 12c (12.2.1.4.0) • Oracle Identity Governance 12c (12.2.1.3.0)	You can use one of the following releases of Oracle Identity Governance or Oracle Identity Manager: • Oracle Identity Governance 12c (12.2.1.4.0) • Oracle Identity Governance 12c (12.2.1.3.0) • Oracle Identity Manager 11*g* Release 2 PS3 (11.1.2.3.0) and any later BP in this release track

Table 1-1 (Cont.) Certified Components

Component	Requirement for AOB Application	Required for CI-Based Connector
Target system	The target system can be any one of the following: • Oracle E-Business Suite 12.1.1 through 12.1.3 • Oracle E-Business Suite 12.2.*x* These applications may run on Oracle Database 10*g*, 11*g*, 12c, or 19c, as either single database or Oracle RAC implementation. **Note:** • If your target system is running on Oracle Database release 19.*x*, then download and apply the Oracle Database patch 31142749 from My Oracle Support. Applying this patch ensures that provisioning operations work fine. • Communication between Oracle Identity Governance and the target system can be in SSL or non-SSL mode.	The target system can be any one of the following: • Oracle E-Business Suite 12.1.1 through 12.1.3 • Oracle E-Business Suite 12.2.*x* These applications may run on Oracle Database 10*g*, 11*g*, 12c, or 19c, as either single database or Oracle RAC implementation. **Note:** • If your target system is running on Oracle Database release 19.*x*, then download and apply the Oracle Database patch 31142749 from My Oracle Support. Applying this patch ensures that provisioning operations work fine. • Communication between Oracle Identity Governance and the target system can be in SSL or non-SSL mode.
Connector server	11.1.2.1.0 or later	11.1.2.1.0 or later
Connector Server JDK	JDK 1.6 or later	JDK 1.6 or later

Table 1-1 (Cont.) Certified Components

Component	Requirement for AOB Application	Required for CI-Based Connector
SSO system	The target system can use one of the following single sign-on (SSO) solutions: • Oracle Single Sign-on with Oracle Internet Directory (release 11.1.1.7.0) as LDAP based repository • Oracle Access Manager with Microsoft Active Directory (2008, 2012 R2), Oracle Directory Server Enterprise Edition (11.1.1.7.0) or Novel eDirectory (8.8) as the LDAP-based repository	The target system can use one of the following single sign-on (SSO) solutions: • Oracle Single Sign-on with Oracle Internet Directory (release 11.1.1.7.0) as LDAP based repository • Oracle Access Manager with Microsoft Active Directory (2008, 2012 R2), Oracle Directory Server Enterprise Edition (11.1.1.7.0) or Novel eDirectory (8.8) as the LDAP-based repository

1.3 Usage Recommendation

These are the recommendations for the EBS User Management connector versions that you can deploy and use depending on the Oracle Identity Governance or Oracle Identity Manager version that you are using.

- If you are using Oracle Identity Governance release 12c (12.2.1.3.0) or later, then use the latest 12.2.1.*x* version of this connector. Deploy the connector using the **Applications** option on the **Manage** tab of Identity Self Service.
- If you are using any of the Oracle Identity Manager releases listed in the "Requirement for CI-Based Connector" column in Certified Components, then use the 11.1.*x* version of the connector. If you want to use the 12.2.1.*x* version of this connector, then you can install and use it only in the CI-based mode. If you want to use the AOB application, then you must upgrade to Oracle Identity Governance release 12c (12.2.1.3.0) or later.

> **Note:**
>
> If you are using the latest 12.2.1.*x* version of the EBS User Management connector in the CI-based mode, then see *Oracle Identity Manager Connector Guide for Oracle E-Business Suite User Management*, Release 11.1.1 for complete details on connector deployment, usage, and customization.

1.4 Certified Languages

These are the languages that the connector supports.

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Czech
- Danish
- Dutch
- English
- Finnish
- French
- French (Canadian)
- German
- Greek
- Hebrew
- Hungarian
- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese
- Portuguese (Brazilian)
- Romanian
- Russian
- Slovak
- Spanish
- Swedish
- Thai
- Turkish

1.5 Supported Connector Operations

These are the list of operations that the connector supports for your target system.

Table 1-2 Supported Connector Operations

Operation	Supported?
User Management	
Create person	Yes
Update person	Yes
Delete person	Yes
Enable person	Yes
Disable person	Yes
Entitlement Grant Management	
Add role	Yes
Update role	Yes
Remove role	Yes
Add responsibility	Yes
Update responsibility	Yes
Remove responsibility	Yes

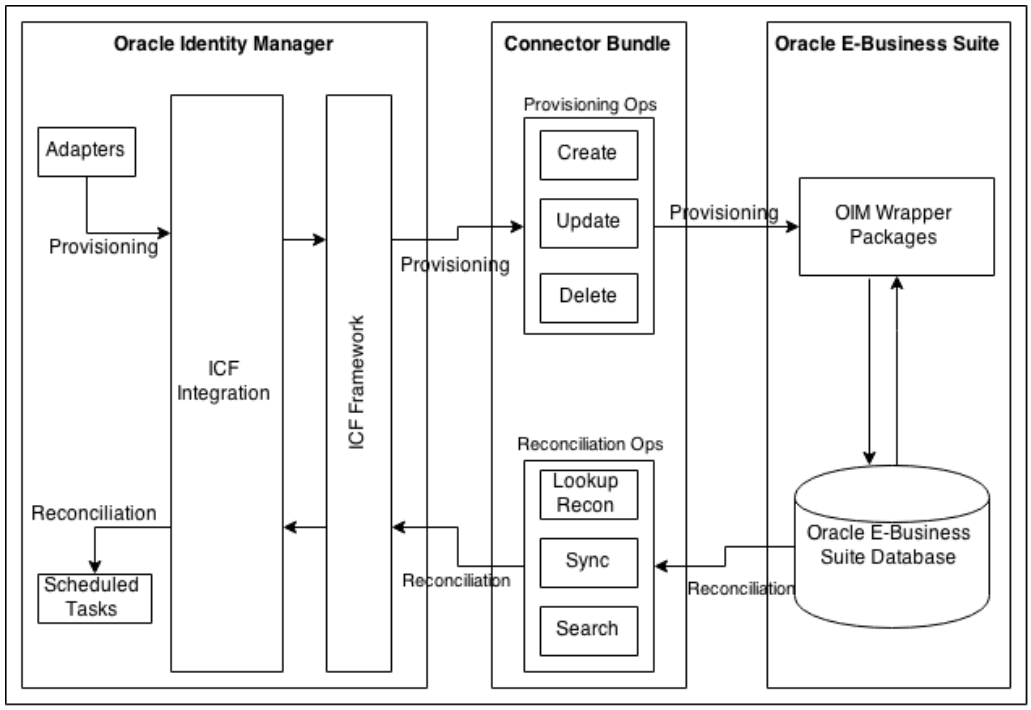
1.6 Connector Architecture

You can configure the Oracle E-Business User Management connector to run in the Target (or account management) mode, and is implemented using the Integrated Common Framework (ICF) component.

The ICF is a component that provides basic reconciliation and provisioning operations that are common to all Oracle Identity Governance connectors. In addition, ICF provides common features that developers would otherwise need to implement on their own, such as connection pooling, buffering, time outs, and filtering. The ICF is shipped along with Oracle Identity Governance. Therefore, you need not configure or modify the ICF.

Figure 1-1 shows the architecture of the Oracle E-Business Suite connectors.

Figure 1-1 Architecture of the Oracle E-Business Suite Connectors



During connector operations, Oracle Identity Governance interacts with a layer called ICF integration. ICF integration is specific to each application with which OIG interacts and uses the ICF API to invoke operations on the Identity Connector (IC). The connector then calls the target system APIs to perform operations on the resource.

The connector communicates with the target system by making calls to the stored procedures in OIG Wrapper packages, which in turn call the target system stored procedures internally. The OIG Wrapper packages are created in the target system when you run a script that is present in the connector installation package. The procedure to run this script is discussed later in this guide.

The basic function of this connector is to enable management of user data on Oracle E-Business Suite through Oracle Identity Governance. In other words, the Oracle E-Business Suite User Management connector enables you to use Oracle E-Business Suite (the target system) as a managed or target resource of Oracle Identity Governance. You can create and manage target system accounts (resources) for OIG Users through provisioning. In addition, data related to newly created and modified target system accounts can be reconciled (using scheduled tasks) and linked with existing OIG Users and provisioned resources.

1.7 Supported Connector Features Matrix

Provides the list of features supported by the AOB application and CI-based connector.

Table 1-3 Supported Connector Features Matrix

Feature	AOB Application	CI-Based Connector
Integrate the target system as a target resource of Oracle Identity Governance	Yes	Yes
Perform Segregation of Duties (SoD) validation of role and responsibility entitlement requests	Yes	Yes
Configure the connector for a single sign-on solution	Yes	Yes
Set account status for reconciliation and provisioning	Yes	Yes
Perform basic password management tasks	Yes	Yes
Perform full and incremental reconciliation	Yes	Yes
Perform limited reconciliation	Yes	Yes
Perform batched reconciliation	Yes	Yes
Configure validation and transformation of account data	Yes	Yes
Install connector in a connector server	Yes	Yes
Use connection pooling	Yes	Yes
Use scheduled jobs for reconciliation of user entities	Yes	Yes
Configure SSL communication between the target system and Oracle Identity Governance	Yes	Yes

1.8 Connector Features

The features of the connector include support for connector server, target resource reconciliation, Segregation of Duties (SoD) validation of role and responsibility entitlement requests, reconciliation of all existing or modified account data, limited and batched reconciliation, transformation and validation of account data during reconciliation and provisioning, and so on.

The following are the features of the connector:

- Support for Target Resource Reconciliation
- SoD Validation of Entitlement Provisioning
- Support for an SSO-Enabled Target System Installation
- Account Status Reconciliation and Provisioning
- Account Password Management
- Full and Incremental Reconciliation

- Support for Batched Reconciliation
- Support for Limited (Filtered) Reconciliation
- Support for Cloning Applications and Creating Instance Applications
- Transformation and Validation of Account Data
- Support for the Connector Server
- Connection Pooling
- Support for SSL Communication Between the Target System and Oracle Identity Governance

1.8.1 Support for Target Resource Reconciliation

You can use the EBS UM connector to configure the target system as a target resource of Oracle Identity Governance.

In this mode, you can use this connector to provision and reconcile the following entities from Oracle E-Business Suite:

- EBS accounts/FND_USR records
- TCA Party records/Vendor records

See Configuring Reconciliation for related information.

1.8.2 SoD Validation of Entitlement Provisioning

This connector supports the SoD feature. Use the Identity Audit (IDA) feature of Oracle Identity Governance to detect SoD violations.

The SoD engine processes role and responsibility entitlement requests that are sent through the connector. Potential conflicts in role and responsibility assignments can be automatically detected.

If you want to enable and use the SoD feature of Oracle Identity Governance with this target system, then you must enable and configure the Identity Audit feature as described in Managing Identity Audit of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

1.8.3 Support for an SSO-Enabled Target System Installation

Oracle E-Business Suite can be configured to use a single sign-on solution such as Oracle Single Sign-On and Oracle Access Manager, to authenticate users. Oracle Single Sign-On uses Oracle Internet Directory as an LDAP-based repository for storing user records. Oracle Access Manager can use Microsoft Active Directory, Oracle Directory Server Enterprise Edition, or Novell eDirectory as the LDAP-based repository.

You can configure the connector to work with either one of these SSO solutions during reconciliation and provisioning operations.

The connector is shipped with an adapter that is responsible for copying SSO account details such as GUID and so on from an enterprise directory process form to EBS user process form.

See Configuring the Connector for SSO for information about configuring the connector for a single sign-on solution.

1.8.4 Account Status Reconciliation and Provisioning

When you enable an account on the target system, the Effective Date From field is set to the current date and the Effective Date To field is set to NULL on the target system.

When you disable an account on the target system, the Effective Date To field is set to the current date on the target system.

The same effect can be achieved through provisioning operations performed on Oracle Identity Governance. In addition, status changes made directly on the target system can be copied into Oracle Identity Governance during reconciliation.

1.8.5 Account Password Management

The connector supports basic password management features. For a particular user, you can specify when the user's password must expire by using the process form fields.

- Password Expiration Type

  You use the Password Expiration Type field to specify the factor (or measure) that you want to use to set a value for password expiration. You can select either `Accesses` or `Days` as the password expiration type.

- Password Expiration Interval

  In the Password Expiration Interval field, you specify the number of access or days for which the user must be able to use the password.

For example, if you specify `Accesses` in the Password Expiration Type field and enter `20` in the Password Expiration Interval field, then the user is prompted to change the user's password at the twenty-first login. Similarly, if you specify `Days` in the Password Expiration Type field and enter `100` in the Password Expiration Interval field, then the user is prompted to change the user's password on the hundred and first day after setting a new password.

See Lookup.Oracle EBS UM.PasswordExpTypes for information about the lookup definition corresponding to the Password Expiration Type field.

1.8.6 Full and Incremental Reconciliation

After you deploy the connector, you can perform full reconciliation to bring all existing user data from the target system to Oracle Identity Governance. After the first full reconciliation run, incremental reconciliation is automatically enabled from the next run of the user reconciliation.

You can perform a full reconciliation run at any time. See Performing Full and Incremental Reconciliation for more information on performing full and incremental reconciliation runs.

1.8.7 Support for Batched Reconciliation

You can break down a reconciliation run into batches by specifying the number of records that must be included in each batch.

See Performing Batched Reconciliation for more information on performing batched reconciliation.

1.8.8 Support for Limited (Filtered) Reconciliation

You can set a reconciliation filter as the value of the Filter attribute of the scheduled tasks. This filter specifies the subset of newly added and modified target system records that must be reconciled.

See Performing Limited Reconciliation for more information on performing limited reconciliation.

1.8.9 Support for Cloning Applications and Creating Instance Applications

You can configure this connector for multiple installations of the target system by cloning applications or by creating instance applications.

When you clone an application, all the configurations of the base application are copied into the cloned application. When you create an instance application, it shares all configurations as the base application.

For more information about these configurations, see Cloning Applications and Creating Instance Applications in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

1.8.10 Transformation and Validation of Account Data

You can configure transformation and validation of account data that is brought into or sent from Oracle Identity Governance during reconciliation and provisioning operations by writing Groovy scripts while creating your application.

For more information, see Validation and Transformation of Provisioning and Reconciliation Attributes in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

1.8.11 Support for the Connector Server

Connector Server is one of the features provided by ICF. By using one or more connector servers, the connector architecture permits your application to communicate with externally deployed bundles.

A Java connector server is useful when you do not wish to execute a Java connector bundle in the same VM as your application. It can be beneficial to run a Java connector on a different host for performance improvements.

For information about installing, configuring, and running the Connector Server, and then installing the connector in a Connector Server, see Using an Identity Connector Server in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

1.8.12 Connection Pooling

A connection pool is a cache of objects that represent physical connections to the target. Oracle Identity Governance connectors can use these connections to communicate with target systems.

At run time, the application requests a connection from the pool. If a connection is available, then the connector uses it and then returns it to the pool. A connection returned to the pool can again be requested for and used by the connector for another operation. By enabling the reuse of connections, the connection pool helps reduce connection creation overheads like network latency, memory allocation, and authentication.

One connection pool is created for each set of basic configuration parameters that you provide while creating an application. For example, if you have three applications for three installations of the target system, then three connection pools will be created, one for each target system installation.

For more information about the parameters that you can configure for connection pooling, see Advanced Settings Parameters .

1.8.13 Support for SSL Communication Between the Target System and Oracle Identity Governance

You can configure SSL to secure communication between Oracle Identity Governance and the target system.

See Configuring Secure Communication Between the Target System and Oracle Identity Governance for more information about securing communication between the target system and Oracle Identity Governance.

2

Creating an Application By Using the EBS User Management Connector

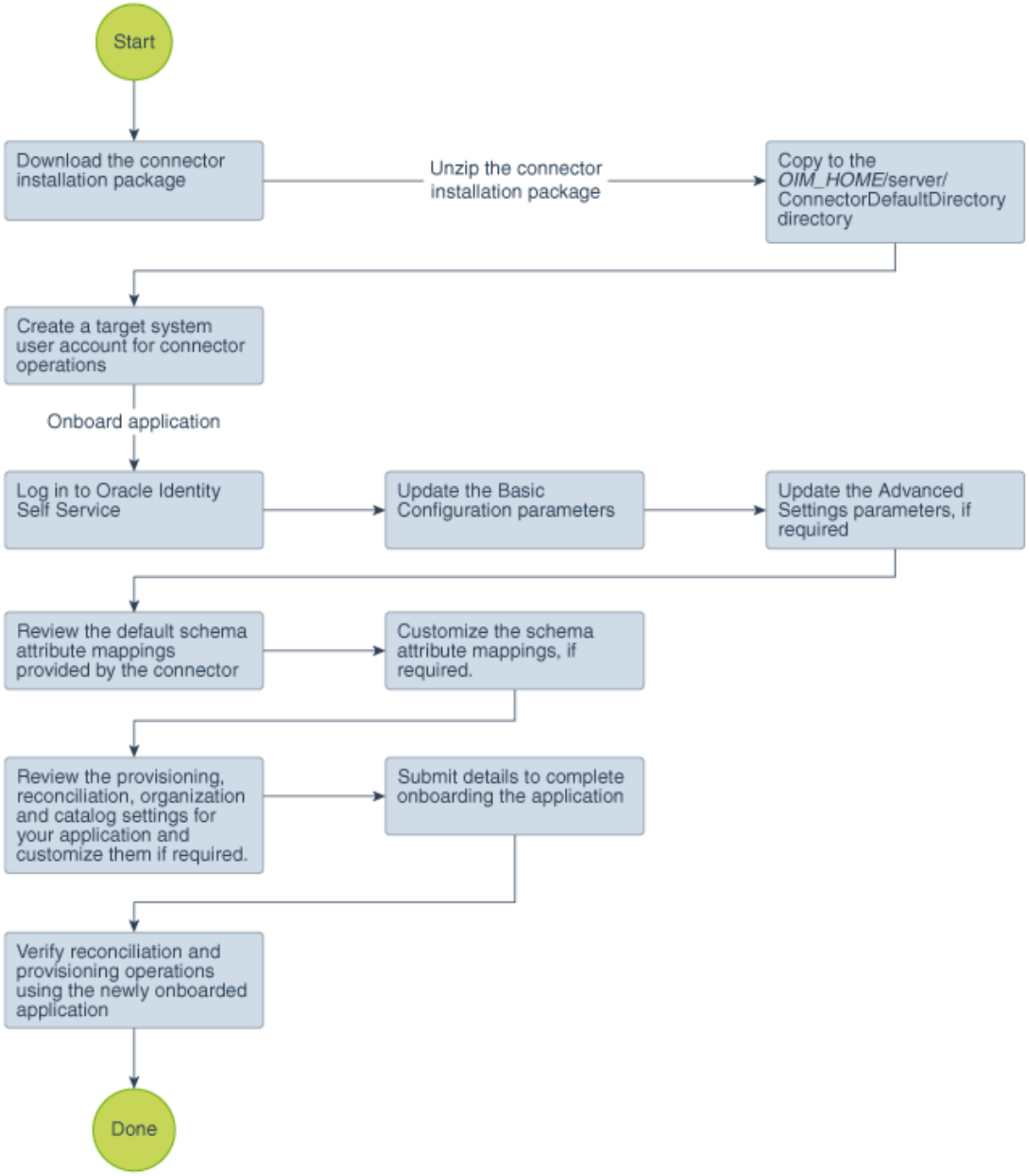
Learn about onboarding applications using the connector and the prerequisites for doing so.

2.1 Process Flow for Creating an Application By Using the Connector

From Oracle Identity Governance release 12.2.1.3.0 onward, connector deployment is handled using the application onboarding capability of Identity Self Service.

Figure 2-1 is a flowchart depicting high-level steps for creating an application in Oracle Identity Governance by using the connector installation package.

Figure 2-1 Overall Flow of the Process for Creating an Application By Using the Connector



2.2 Prerequisites for Creating an Application By Using the Connector

Learn about the tasks that you must complete before you create the application.

- Downloading the Connector Installation Package
- Creating a Target System User Account for Connector Operations
- Determining Values for the JDBC URL and Connection Properties Parameters

2.2.1 Downloading the Connector Installation Package

You can obtain the installation package for your connector on the Oracle Technology Network (OTN) website.

To download the connector installation package:

1. Navigate to the OTN website at http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html.
2. Click **OTN License Agreement** and read the license agreement.
3. Select the **Accept License Agreement** option.

   You must accept the license agreement before you can download the installation package.
4. Download and save the installation package to any directory on the computer hosting Oracle Identity Governance.
5. Extract the contents of the installation package to any directory on the computer hosting Oracle Identity Governance. This creates a directory named *CONNECTOR_NAME-RELEASE_NUMBER.*
6. Copy the *CONNECTOR_NAME-RELEASE_NUMBER* directory to the *OIG_HOME*/server/ConnectorDefaultDirectory directory.

2.2.2 Creating a Target System User Account for Connector Operations

This preinstallation step involves creating a user account in the target system that can be used by the connector to perform connector operations.

> **Note:**
>
> You must have DBA privileges to run the scripts described in this section and grant the required permissions to the target system user account.
>
> You must have Oracle Database Client installed on the computer on which you perform the procedure described in this section. The Oracle Database Client release must be the same as the database release. In addition, if Oracle Database Client is not installed on the database host computer, then the tnsnames.ora file on the Oracle Database Client host must contain an entry for the SID of the database.

Oracle Identity Governance requires a target system user account to access the target system during connector operations. You provide the credentials of this user account as part of Basic Configuration Parameters while creating an application.

To create a target system user account for connector operations:

1. From the installation media, copy the scripts directory to a temporary directory on either the target system host computer or a computer on which the Oracle Database Client has been installed.
2. On the computer where you copy the scripts directory, verify that there is a TNS entry in the tnsnames.ora file for the target system database.
3. Change to the directory containing the scripts directory and depending on the host platform, run either the Run_UM_DBScripts.sh or Run_UM_DBScripts.bat file. These files are present in the scripts directory of the installation media.
4. When you run the script, you are prompted for the following information:
   - `Enter the ORACLE_HOME`

     Set a value for the ORACLE_HOME environment variable. This prompt is displayed only if the ORACLE_HOME environment variable has not been set on the computer on which you are running the script.
   - `Enter the System User Name`

     Enter the login (user name) of a DBA account with the privileges to create and configure a new target system user.
   - `Enter the name of the database`

     Enter the connection string or service name given in the tnsnames.ora file to connect to the target system database.

     This connects you to the SQL*Plus client.
   - `Enter password`

Enter the password of the APPS user in the target system. The Type and Package are created, and then the connection to the database is disconnected.

- `Enter password`

  Enter the password of the dba user.

- `Enter New database Username to be created`

  Enter a user name for the target system account that you want to create.

- `Enter the New user password`

  Enter a password for the target system account that you want to create.

  This installs all wrappers packages under the APPS schema, creates the new target system account, and then grants all the required privileges on the tables and packages.

- `Connecting with newly created database user`

  Enter the connection string or service name that you provided earlier.

The user account for connector operations is created.

2.2.3 Determining Values for the JDBC URL and Connection Properties Parameters

This section discusses the JDBC URL and Connection Properties parameters. You apply the information in this section while configuring the IT resource for your target system. This procedure is discussed later in this guide.

The values that you specify for the JDBC URL and Connection Properties parameters depend on the security measures that you have implemented:

- Supported JDBC URL Formats
- Only SSL Communication Is Configured
- Both Data Encryption and Integrity and SSL Communication Are Configured

2.2.3.1 Supported JDBC URL Formats

The following are the supported JDBC URL formats:

- Multiple database instances support one service (Oracle RAC)

  JDBC URL format:

  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
  (HOST=HOST1_NAME.DOMAIN)(PORT=PORT1_NUMBER))(ADDRESS=(PROTOCOL=TCP)
  (HOST=HOST2_NAME.DOMAIN)(PORT=PORT2_NUMBER))(ADDRESS=(PROTOCOL=TCP)
  (HOST=HOST3_NAME.DOMAIN)(PORT=PORT3_NUMBER)) . . .
  (ADDRESS=(PROTOCOL=TCP)(HOST=HOSTn_NAME.DOMAIN)(PORT=PORTn_NUMBER))
  (CONNECT_DATA=(SERVICE_NAME=ORACLE_DATABASE_SERVICE_NAME)))
  ```

  Sample value:

  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=
  host1.example.com)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=
  host2.example.com)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=
  ```

```
host3.example.com)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=
host4.example.com)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME= srvce1)))
```

- One database instance supports one service

  JDBC URL format:

  `jdbc:oracle:thin:@`*`HOST_NAME.DOMAIN:PORT_NUMBER:ORACLE_DATABASE_SERVICE_NAME`*

  Sample value:

  `jdbc:oracle:thin:@host1.example:1521:srvce1`

- One database instance supports multiple services (for Oracle Database 10*g* and later)

  JDBC URL format:

  ```
  jdbc:oracle:thin:@//HOST_NAME.DOMAIN:PORT_NUMBER/
  ORACLE_DATABASE_SERVICE_NAME
  ```

  Sample value:

  `jdbc:oracle:thin:@host1.example.com:1521/srvce1`

2.2.3.2 Only SSL Communication Is Configured

After you configure SSL communication, the database URL is recorded in the tnsnames.ora file. See Local Naming Parameters in the tnsnames.ora File in *Oracle Database Net Services Reference* for detailed information about the tnsnames.ora file.

The following are sample formats of the contents of the tnsnames.ora file. In these formats, `DESCRIPTION` contains the connection descriptor, `ADDRESS` contains the protocol address, and `CONNECT_DATA` contains the database service identification information.

Sample Format 1:

```
NET_SERVICE_NAME=
 (DESCRIPTION=
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (CONNECT_DATA=
     (SERVICE_NAME=SERVICE_NAME)))
```

Sample Format 2:

```
NET_SERVICE_NAME=
 (DESCRIPTION_LIST=
  (DESCRIPTION=
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (CONNECT_DATA=
     (SERVICE_NAME=SERVICE_NAME)))
  (DESCRIPTION=
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (CONNECT_DATA=
     (SERVICE_NAME=SERVICE_NAME))))
```

Sample Format 3:

```
NET_SERVICE_NAME=
 (DESCRIPTION=
  (ADDRESS_LIST=
   (LOAD_BALANCE=on)
   (FAILOVER=off)
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION)))
  (ADDRESS_LIST=
   (LOAD_BALANCE=off)
   (FAILOVER=on)
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
   (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION)))
  (CONNECT_DATA=
   (SERVICE_NAME=SERVICE_NAME)))
```

If you have configured only SSL communication and imported the certificate that you create on the target system host computer into the JVM certificate store of Oracle Identity Manager, then you must derive the value for the JDBC URL parameter from the value of *NET_SERVICE_NAME* in the tnsnames.ora file. For example:

> **Note:**
>
> As shown in this example, you must include only the (ADDRESS=(PROTOCOL=TCPS)(HOST=*HOST_NAME*)(PORT=2484)) element because you are configuring SSL. You need not include other (ADDRESS=(*PROTOCOL_ADDRESS_INFORMATION*)) elements.

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS)
(HOST=myhost)(PORT=2484)))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=mysid)))
```

2.2.3.3 Both Data Encryption and Integrity and SSL Communication Are Configured

If both data encryption and integrity and SSL communication are configured, then specify a value for the JDBC URL parameter in the following manner:

Enter a comma-separated combination of the values for the JDBC URL parameter described in Only SSL Communication Is Configured. For example:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS)
(HOST=myhost)(PORT=2484)))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=mysid)))
```

2.3 Creating an Application By Using the Connector

You can onboard an application into Oracle Identity Governance from the connector package by creating a target application. To do so, you must log in to Identity Self Service and then choose the **Applications** box on the **Manage** tab.

The following is the high-level procedure to create an application by using the connector:

> **Note:**
>
> For detailed information on each of the steps in this procedure, see Creating Applications of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

1. Create an application in Identity Self Service. The high-level steps are as follows:
   a. Log in to Identity Self Service either by using the **System Administration** account or an account with the **ApplicationInstanceAdministrator** admin role.
   b. Ensure that the **Connector Package** option is selected when creating an application.
   c. Update the basic configuration parameters to include connectivity-related information.
   d. If required, update the advanced setting parameters to update configuration entries related to connector operations.
   e. Review the default user account attribute mappings. If required, add new attributes or you can edit or delete existing attributes.
   f. Review the provisioning, reconciliation, organization, and catalog settings for your application and customize them if required. For example, you can customize the default correlation rules for your application if required.
   g. Review the details of the application and click **Finish** to submit the application details.

      The application is created in Oracle Identity Governance.
   h. When you are prompted whether you want to create a default request form, click **Yes** or **No**.

      If you click **Yes**, then the default form is automatically created and is attached with the newly created application. The default form is created with the same name as the application. The default form cannot be modified later. Therefore, if you want to customize it, click **No** to manually create a new form and attach it with your application.
2. Verify reconciliation and provisioning operations on the newly created application.

> **See Also:**
>
> - Configuring the EBS User Management Connector for details on basic configuration and advanced settings parameters, default user account attribute mappings, default correlation rules, and reconciliation jobs that are predefined for this connector
> - Configuring Oracle Identity Governance for details on creating a new form and associating it with your application, if you chose not to create the default form

3 Configuring the EBS User Management Connector

While creating an application, you must configure connection-related parameters that the connector uses to connect Oracle Identity Governance with your target system and perform connector operations. In addition, you can view and edit attribute mappings between the process form fields in Oracle Identity Governance and target system columns, predefined correlation rules, situations and responses, and reconciliation jobs.

3.1 Basic Configuration Parameters

These are the connection-related parameters that Oracle Identity Governance requires to connect to the EBS User Management connector.

Table 3-1 Basic Configuration Parameters for the Connector

Parameter	Mandatory?	Description
Connection URL	Yes	Enter the database connection string using the `host:post:sid` syntax format. Sample value: `jdbc:oracle:thin:@%host:%port:%sid` See Determining Values for the JDBC URL and Connection Properties Parameters for information about JDBC URL formats.
User	Yes	Enter the user ID of the database user account that Oracle Identity Governance uses to connect to the target system. Sample value: `sys as sysdba`

Table 3-1 (Cont.) Basic Configuration Parameters for the Connector

Parameter	Mandatory?	Description
Password	Yes	Enter the password for the user name of the target system account to be used for connector operations.
Connector Server Name	No	If you created an IT resource of the type “Connector Server”, then enter its name.
Topology Name	No	Enter the name of the SoD topology, if any SoD integration exists. The value must be the same as the value of the topologyName element in the SILConfig.xml file. If you are using default SIL registration, then specify `sodoaacg` as the value. Default value: `None` **Note:** The Topology Name parameter is *deprecated* as SoD violations are detected using the Identity Audit feature. For more information about enabling and configuring the Identity Audit feature, see Managing Identity Audit in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.
Batch Size	No	Enter the number of records that must be included in each batch fetched from the target system during reconciliation. Default value: `1000`
Context App ID	No	An application context is a set of elements associated with an artifact in Oracle E-Business Suite. The context implements user preferences and access control on the artifact. The Context App ID, Context Resp ID, and Context User ID parameters define the context that is used for connector operations. Enter the name of the application to which the user belongs. Default value: `0`

Table 3-1 (Cont.) Basic Configuration Parameters for the Connector

Parameter	Mandatory?	Description
Context Resp ID	No	Enter the responsibility assigned to the user in whose context connector operations are performed on the target system. Default value: `0`
Context User ID	No	Enter the user ID of the user in whose context connector operations are performed on the target system. Default value: `0`
Database	No	Enter the name of the target system database.
Host	No	Enter the host name or IP address of the computer hosting the target system.
Port	No	Enter the number of the port at which the target system database is listening.

3.2 Advanced Settings Parameters

These are the configuration-related entries that the connector uses during reconciliation and provisioning operations.

Table 3-2 Advanced Settings Parameters for the Connector

Parameter	Mandatory?	Description
Connector Name	Yes	This parameter holds the name of the connector class. **Default Value:** `org.identityconnectors.ebs.EBSConnector`
Bundle Name	Yes	This parameter holds the name of the connector bundle package. **Default Value:** `org.identityconnectors.ebs`
Bundle Version	Yes	This parameter holds the version of the connector bundle class. **Default Value:** `12.3.0`
Pool Max Idle	No	Maximum number of idle objects in a pool. **Default value**: 10

Table 3-2 (Cont.) Advanced Settings Parameters for the Connector

Parameter	Mandatory?	Description
Pool Max Size	No	Maximum number of connections that the pool can create. **Default value**: 10
Pool Max Wait	No	Maximum time, in milliseconds, the pool must wait for a free object to make itself available to be consumed for an operation. **Default value**: 150000
Pool Min Evict Idle Time	No	Minimum time, in milliseconds, the connector must wait before evicting an idle object. **Default value**: 120000
Pool Min Idle	No	Minimum number of idle objects in a pool. **Default value**: 1
FilterDateAttributes	No	Date attributes used for filtering the user. If you want to add more date fields, then you need to provide all those date field names with comma separator in decode value. **Default value**: `START_DATE`
FilterDateAttributeFormat	No	Date attribute value format. **Default value**: `dd-MMM-yyyy`

3.3 Attribute Mappings

The Schema page for a target application displays the default schema (provided by the connector) that maps Oracle Identity Governance attributes to target system columns. The connector uses these mappings during reconciliation and provisioning operations.

Oracle EBS UM User Account Attributes

Table 3-3 lists the user-specific attribute mappings between the process form fields in Oracle Identity Governance and EBS User Management columns. The table also lists whether a specific attribute is used during provisioning or reconciliation and whether it is a matching key field for fetching records during reconciliation.

If required, you can edit the default attribute mappings by adding new attributes or deleting existing attributes as described in Creating a Target Application in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-3 Default Attribute Mappings for Oracle EBS UM User Account

Display Name	Target Attribute	Data Type	Mandatory Provisioning Property?	Provision Field?	Recon Field	Key Field?	Case Insensitive?
Party Last Name	PARTY_LAST_NAME	String	No	Yes	Yes	No	Not applicable
Description	DESCRIPTION	String	No	Yes	Yes	No	Not applicable
Person Id	EMPLOYEE_ID	String	No	Yes	Yes	No	Not applicable
Effective Start Date	START_DATE	Date	No	Yes	Yes	No	Not applicable
Supplier Name	SUPPLIER_NAME	String	No	Yes	Yes	No	Not applicable
Fax	FAX	String	No	Yes	Yes	No	Not applicable
Effective End Date	END_DATE	Date	No	Yes	Yes	No	Not applicable
Password Expiration Type	PASSWORD_EXP_TYPE	String	No	Yes	Yes	No	Not applicable
Party Type	PARTY_TYPE	String	No	Yes	Yes	No	Not applicable
Party Id	PARTY_ID	String	No	Yes	Yes	No	Not applicable
User Name	__NAME__	String	Yes	Yes	Yes	No	Not applicable
Party First Name	PARTY_FIRST_NAME	String	No	Yes	Yes	No	Not applicable
Email	EMAIL_ADDRESS	String	No	Yes	Yes	No	Not applicable
SSO GUID	USER_GUID	String	No	Yes	Yes	No	Not applicable
Password Expiration Interval	PASSWORD_LIFESPAN	Long	No	Yes	Yes	No	Not applicable
Supplier Party Id	SUPPLIER_PARTY_ID	String	No	No	Yes	No	Not applicable
User Id	__UID__	String	No	No	Yes	Yes	No
Status	__ENABLE__	String	No	No	Yes	No	Not applicable

Table 3-3 (Cont.) Default Attribute Mappings for Oracle EBS UM User Account

Display Name	Target Attribute	Data Type	Mandatory Provisioning Property?	Provision Field?	Recon Field	Key Field?	Case Insensitive?
IT Resource Name	NA	Long	No	No	Yes	No	Not applicable
Password	__PASSWORD__	String	No	Yes	No	No	Not applicable
SoDCheck TrackingID	NA	String	No	No	No	No	Not applicable
SoDCheck Timestamp	NA	String	No	No	No	No	Not applicable
SoDCheck EntitlementViolation	NA	String	No	No	No	No	Not applicable
SoDCheck Status	NA	Date	No	No	No	No	Not applicable
SoDCheck Result	NA	String	No	No	No	No	Not applicable

Figure 3-1 shows the default User account attribute mappings.

Figure 3-1 Default Attribute Mappings for Oracle EBS UM User Account

Oracle EBS UM User

Add Attribute

Application Attribute				Provisioning Property		Reconciliation Properties			
Identity Attribute	Display Name	Target Attribute	Data Type	Mandatory	Provision Field	Recon Field	Key Field	Case Insensitive	
Select a value	Party Last Name	PARTY_LAST_NAME	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	Description	DESCRIPTION	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	Person Id	EMPLOYEE_ID	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	Effective Start Dat	START_DATE	Date	☐	☑	☑	☐	☐	✖ ☰
Select a value	User Id	__UID__	String	☐	☐	☑	☑	☐	✖ ☰
Select a value	Supplier Name	SUPPLIER_NAME	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	SoDCheckTrackin		String	☐	☐	☐	☐	☐	✖ ☰
Select a value	SoDCheckTimesta		String	☐	☐	☐	☐	☐	✖ ☰
Select a value	SoDCheckEntitlen		String	☐	☐	☐	☐	☐	✖ ☰
Select a value	Fax	FAX	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	Supplier Party Id	SUPPLIER_PARTY_ID	String	☐	☐	☑	☐	☐	✖ ☰
Select a value	Effective End Date	END_DATE	Date	☐	☑	☑	☐	☐	✖ ☰
Select a value	Password Expirati	PASSWORD_EXP_TYPE	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	Party Type	PARTY_TYPE	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	SoDCheckStatus		String	☐	☐	☐	☐	☐	✖ ☰
Select a value	Party Id	PARTY_ID	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	User Name	__NAME__	String	☑	☑	☑	☐	☐	✖ ☰
Select a value	Party First Name	PARTY_FIRST_NAME	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	Password	__PASSWORD__	String	☐	☑	☐	☐	☐	✖ ☰
Select a value	Email	EMAIL_ADDRESS	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	SSO GUID	USER_GUID	String	☐	☑	☑	☐	☐	✖ ☰
Select a value	Password Expirati	PASSWORD_LIFESPAN	Long	☐	☑	☑	☐	☐	✖ ☰
Select a value	SoDCheckResult		String	☐	☐	☐	☐	☐	✖ ☰
Select a value	Status	__ENABLE__	String	☐	☐	☑	☐	☐	✖ ☰
Select a value	IT Resource Name		Long	☐	☐	☑	☐	☐	✖ ☰

Responsibilities Entitlement Attributes

Table 3-4 lists the responsibilities-specific attribute mappings between the process form fields in Oracle Identity Governance and Oracle EBS User Management columns. The table lists whether a given attribute is mandatory during provisioning. It also lists whether a given attribute is used during reconciliation and whether it is a matching key field for fetching records during reconciliation.

If required, you can edit the default attribute mappings by adding new attributes or deleting existing attributes as described in Creating a Target Application in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-4 Default Attribute Mappings for Responsibilities Entitlement

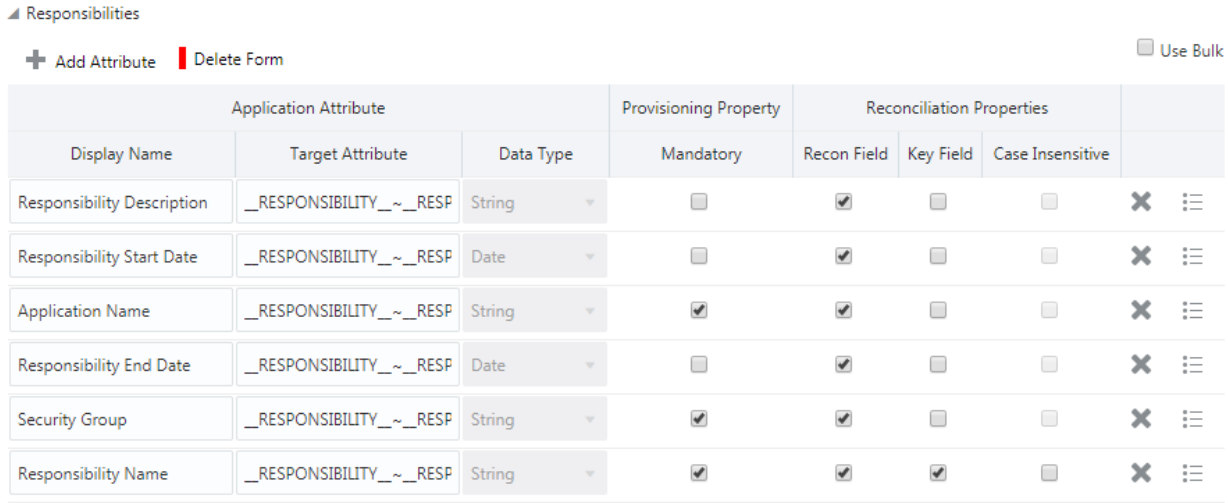
Display Name	Target Attribute	Data Type	Mandatory Provisioning Property?	Recon Field	Key Field?	Case Insensitive?
Application Name	__RESPONSIBILITY__~__RESPONSIBILITY__~RESPONSIBILITY_APP_ID	String	Yes	Yes	No	Not applicable
Security Group	__RESPONSIBILITY__~__RESPONSIBILITY__~SECURITY_GROUP_ID	String	Yes	Yes	No	Not applicable
Responsibility Name	__RESPONSIBILITY__~__RESPONSIBILITY__~RESPONSIBILITY_ID	String	Yes	Yes	Yes	No
Responsibility Description	__RESPONSIBILITY__~__RESPONSIBILITY__~RESP_DESCRIPTION	String	No	Yes	No	Not applicable
Responsibility Start Date	__RESPONSIBILITY__~__RESPONSIBILITY__~RESP_START_DATE	Date	No	Yes	No	Not applicable

Table 3-4 (Cont.) Default Attribute Mappings for Responsibilities Entitlement

Display Name	Target Attribute	Data Type	Mandatory Provisioning Property?	Recon Field	Key Field?	Case Insensitive?
Responsibility End Date	__RESPONSIBILITY__~__RESPONSIBILITY__~RESP_END_DATE	Date	No	Yes	No	Not applicable

Figure 3-2 shows the default Responsibilities entitlement mapping.

Figure 3-2 Default Attribute Mappings for Responsibilities Entitlement



Roles Entitlement Attributes

Table 3-5 lists the roles-specific attribute mappings between the process form fields in Oracle Identity Governance and Oracle EBS User Management columns. The table lists whether a given attribute is mandatory during provisioning. It also lists whether a given attribute is used during reconciliation and whether it is a matching key field for fetching records during reconciliation.

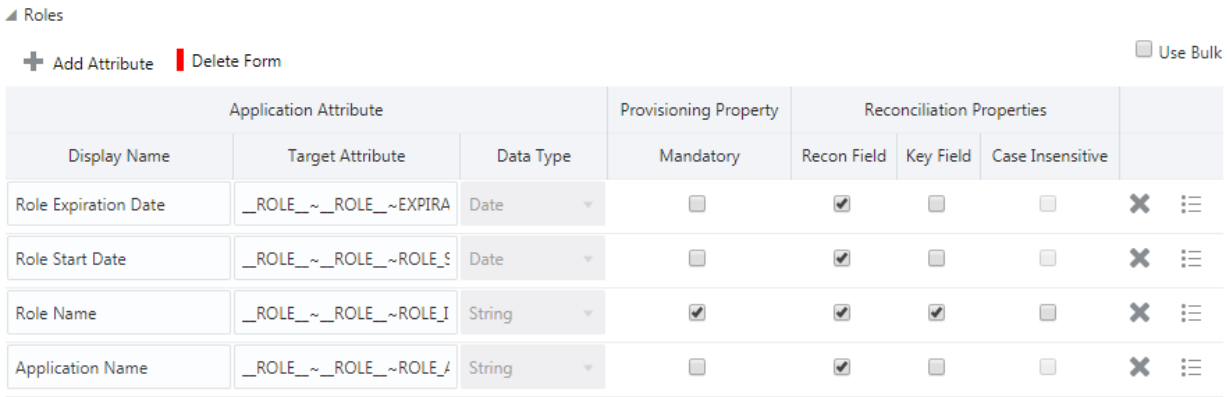
If required, you can edit the default attribute mappings by adding new attributes or deleting existing attributes as described in Creating a Target Application in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-5 Default Attribute Mappings for Roles Entitlement

Display Name	Target Attribute	Data Type	Mandatory Provisioning Property?	Recon Field	Key Field?	Case Insensitive?
Role Name	__ROLE__~__ROLE__~ROLE_ID	String	Yes	Yes	Yes	No
Role Expiration Date	__ROLE__~__ROLE__~EXPIRATION_DATE	Date	No	Yes	No	Not applicable
Role Start Date	__ROLE__~__ROLE__~ROLE_START_DATE	Date	No	Yes	No	Not applicable
Application Name	__ROLE__~__ROLE__~ROLE_APP_ID	String	No	Yes	No	Not applicable

Figure 3-3 shows the default Roles entitlement mapping.

Figure 3-3 Default Attribute Mappings for Roles Entitlement



3.4 Rules, Situations, and Responses

Learn about the predefined rules, responses and situations for a Target application. The connector use these rules and responses for performing reconciliation.

Predefined Identity Correlation Rules

By default, the EBS User Management connector provides a simple correlation rule when you create a Target application. The connector uses this correlation rule to compare the entries in Oracle Identity Governance repository and the target system repository, determine the difference between the two repositories, and apply the latest changes to Oracle Identity Governance.

Table 3-6 lists the default simple correlation rule for the EBS User Management connector. If required, you can edit the default correlation rule or add new rules. You can create complex correlation rules also. For more information about adding or editing simple or complex correlation rules, see Creating a Target Application in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Table 3-6 Predefined Identity Correlation Rule for the EBS User Management Connector

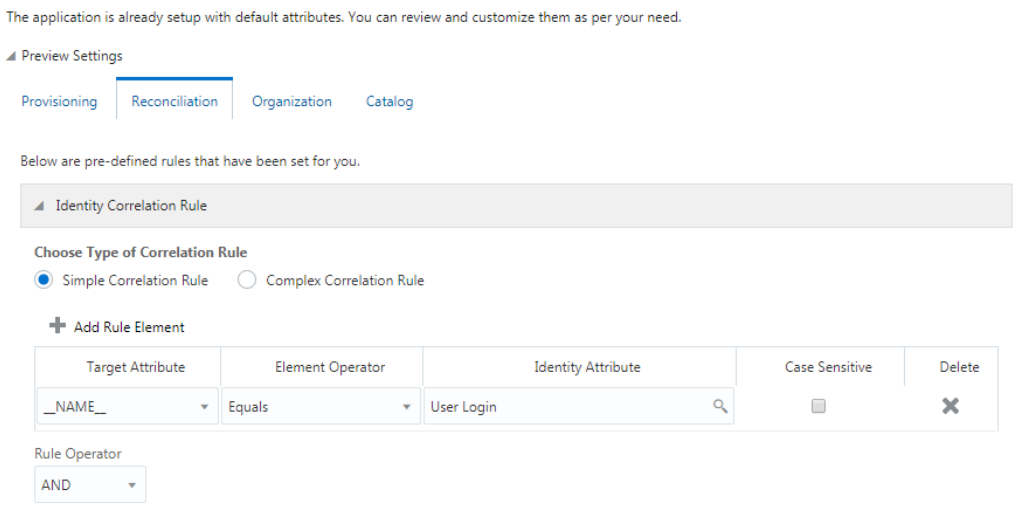
Target Attribute	Element Operator	Identity Attribute	Case Sensitive?
__NAME__	Equals	User Login	No

In this identity rule:

- __NAME__ is a single-valued attribute on the target system that identifies the user account.
- User Login is the field on the OIG User form.

Figure 3-4 shows the simple correlation rule for the EBS User Management Connector.

Figure 3-4 Simple Correlation Rule for the EBS User Management Connector



Predefined Situations and Responses

The EBS User Management connector provides a default set of situations and responses when you create a Target application. These situations and responses specify the action that Oracle Identity Governance must take based on the result of a reconciliation event.

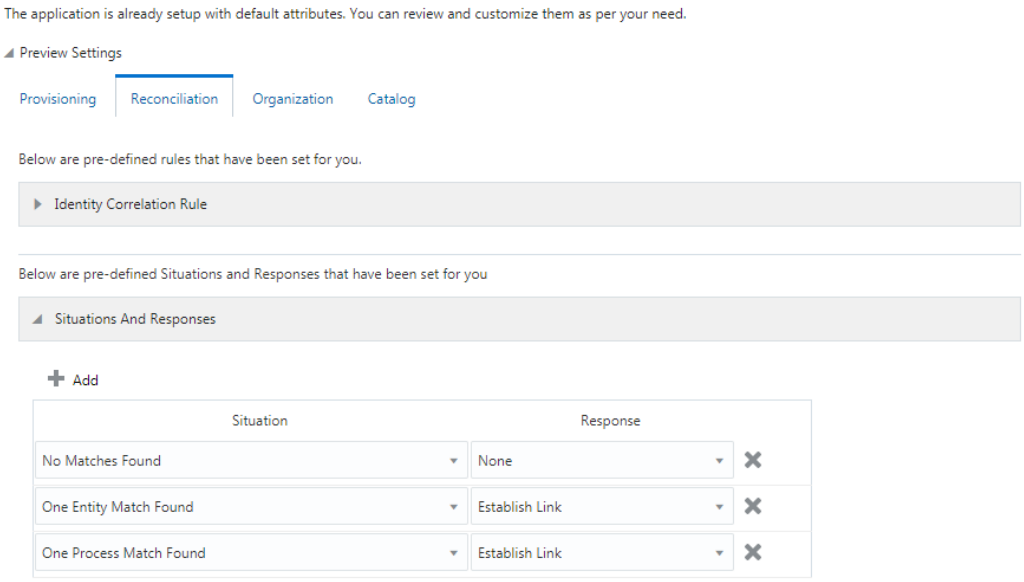
Table 3-7 lists the default situations and responses for the EBS User Management connector. If required, you can edit these default situations and responses or add new ones. For more information about adding or editing situations and responses, see Creating a Target Application in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*

Table 3-7 Predefined Situations and Responses for the EBS User Management Connector

Situation	Response
No Matches Found	None
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

Figure 3-5 shows the situations and responses that the connector provides by default.

Figure 3-5 Predefined Situations and Responses for the EBS User Management Connector



3.5 Reconciliation Jobs

These are the reconciliation jobs that are automatically created in Oracle Identity Governance after you create the application for your target system.

You can either use these predefined jobs or edit them to meet your requirements. Alternatively, you can create custom reconciliation jobs. For information about editing these predefined jobs or creating new ones, see Updating Reconciliation Jobs in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

Full User Reconciliation Job

The Oracle EBS UM Target User Reconciliation job is used to fetch all user records from the target system.

Table 3-8 Parameters of the Oracle EBS UM Target User Reconciliation Job

Parameter	Description
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.
Filter	Enter the expression for filtering records that the scheduled job must reconcile. Sample value: `equalTo('__UID__','SEPT12USER1')` For information about the filters expressions that you can create and use, see ICF Filter Syntax in *Developing and Customizing Applications for Oracle Identity Governance*.
Incremental Recon Attribute	Name of the target system column that holds holds the timestamp at which the user record was modified. Sample value: `lastModified`
Object Type	Type of object you want to reconcile. Default value: `User`
Latest Token	The parameter holds the value of the target system column that is specified as the value of the Incremental Recon Attribute parameter. The Latest Token parameter is used for internal purposes. By default, this value is empty. **Note:** Do not enter a value for this attribute. The reconciliation engine automatically enters a value in this attribute.
Scheduled Task Name	Name of the scheduled job. **Note:** For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a new job or create a copy of the job, then enter the unique name for that scheduled job as the value of this attribute.

Incremental User Reconciliation Job

The Oracle EBS UM Target Incremental User Reconciliation job is used to fetch the records that are added or modified after the last reconciliation run.

Table 3-9 Parameters of the Oracle EBS UM Target Incremental User Reconciliation Job

Parameter	Description
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.

Table 3-9 (Cont.) Parameters of the Oracle EBS UM Target Incremental User Reconciliation Job

Parameter	Description
Sync Token	Enter the expression for filtering records that the scheduled job must reconcile. Sample value: `equalTo('__UID__','SEPT12USER1')` For information about the filters expressions that you can create and use, see ICF Filter Syntax in *Developing and Customizing Applications for Oracle Identity Governance*.
Object Type	Type of object you want to reconcile. Default value: `User`
Scheduled Task Name	Name of the scheduled job. **Note:** For the scheduled job included with this connector, you must not change the value of this attribute. However, if you create a new job or create a copy of the job, then enter the unique name for that scheduled job as the value of this attribute.

Delete User Reconciliation Job

The Oracle EBS UM Target User Delete Reconciliation job is used to reconcile user data when for target application.

Table 3-10 Parameters of the Oracle EBS UM Target User Delete Reconciliation Job

Parameter	Description
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.
Object Type	Type of object you want to reconcile. Default value: `User`

Reconciliation Jobs for Entitlements

The following jobs are available for reconciling entitlements:

- Oracle EBS UM Target Roles Lookup Reconciliation
- Oracle EBS UM Target Responsibilities Lookup Reconciliation
- Oracle EBS UM Target Applications Lookup Reconciliation
- Oracle EBS UM Target Security Groups Lookup Reconciliation

The parameters for all the reconciliation jobs are the same.

Table 3-11 Parameters of the Reconciliation Jobs for Entitlements

Parameter	Description
Application Name	Name of the application you created for your target system. This value is the same as the value that you provided for the Application Name field while creating your target application. Do not modify this value.
Lookup Name	This parameter holds the name of the lookup definition that maps each lookup definition with the data source from which values must be fetched. Depending on the reconciliation job you are using, the default values are as follows: • For Oracle EBS UM Target Roles Lookup Reconciliation: `Lookup.Oracle EBS UM.Roles` • For Oracle EBS UM Target Responsibilities Lookup Reconciliation: `Lookup.Oracle EBS UM.Responsibilities` • For Oracle EBS UM Target Applications Lookup Reconciliation: `Lookup.Oracle EBS UM.Applications` • For Oracle EBS UM Target Security Groups Lookup Reconciliation: `Lookup.Oracle EBS UM.SecurityGroups`
Object Type	Enter the type of object whose values must be synchronized. Depending on the scheduled job you are using, the default values are as follows: • For Oracle EBS UM Target Roles Lookup Reconciliation: `__ROLES__` • For Oracle EBS UM Target Responsibilities Lookup Reconciliation: `__RESPONSIBILITIES__` • For Oracle EBS UM Target Applications Lookup Reconciliation: `__APPLICATIONS__` • For Oracle EBS UM Target Security Groups Lookup Reconciliation: `__SECURITY_GROUPS__` **Note:** Do not change the value of this attribute.
Code Key Attribute	Enter the name of the connector or target system attribute that is used to populate the Code Key column of the lookup definition (specified as the value of the Lookup Name attribute). Default value: `Code` **Note:** Do not change the value of this attribute.
Decode Attribute	Enter the name of the connector or target system attribute that is used to populate the Decode column of the lookup definition (specified as the value of the Lookup Name attribute). Default value: `Decode`

4

Performing the Postconfiguration Tasks for the EBS User Management Connector

These are the tasks that you must perform after creating an application in Oracle Identity Governance.

- Configuring Secure Communication Between the Target System and Oracle Identity Governance
- Configuring Oracle Identity Governance
- Harvesting Entitlements and Sync Catalog
- Managing Logging
- Configuring the Connector for SSO
- Localizing Field Labels in UI Forms

4.1 Configuring Secure Communication Between the Target System and Oracle Identity Governance

To secure communication between Oracle Database and Oracle Identity Governance, you can perform either one or both of the following procedures:

> **Note:**
>
> To perform the procedures described in this section, you must have the permissions required to modify the TNS listener configuration file.

- Configuring Data Encryption and Integrity in Oracle Database
- Configuring SSL Communication in Oracle Database

4.1.1 Configuring Data Encryption and Integrity in Oracle Database

See Data Encryption in *Oracle Database Advanced Security Administrator's Guide* for information about configuring data encryption and integrity.

4.1.2 Configuring SSL Communication in Oracle Database

To enable SSL communication between Oracle Database and Oracle Identity Governance:

1. See Secure Socket Layer in *Oracle Database Advanced Security Administrator's Guide* for information about enabling SSL communication between Oracle Database and Oracle Identity Governance.

2. Export the certificate on the Oracle Database host computer.
3. Copy the certificate to Oracle Identity Governance.
4. Import the certificate into the JVM certificate store of the application server on which Oracle Identity Governance is running.

   To import the certificate into the certificate store, run the following command:

   ```
   keytool -import -file FILE_LOCATION -keystore TRUSTSTORE_LOCATION -storepass
   TRUSTSTORE_PASSWORD -trustcacerts -alias ALIAS
   ```

   In this command:

   - Replace *FILE_LOCATION* with the full path and name of the certificate file.
   - Replace *ALIAS* with an alias for the certificate.
   - Replace *TRUSTSTORE_PASSWORD* with a password for the certificate store.
   - Replace *TRUSTSTORE_LOCATION* with one of the certificate store paths given in Table 4-1. This table shows the location of the certificate store for each of the supported application servers.

> **Note:**
>
> In an Oracle Identity Governance cluster, you must import the file into the certificate store on each node of the cluster.

Table 4-1 Certificate Store Locations

Application Server	Certificate Store Location
Oracle WebLogic Server	• If you are using Oracle jrockit_R27.3.1-jdk, then copy the certificate into the following directory: *JROCKIT_HOME*/jre/lib/security • If you are using the default Oracle WebLogic Server JDK, then copy the certificate into the following directory: *WEBLOGIC_HOME*/java/jre/lib/security/cacerts
IBM WebSphere Application Server	• For a nonclustered configuration of any supported IBM WebSphere Application Server release, import the certificate into the following certificate store: *WEBSPHERE_HOME*/java/jre/lib/security/cacerts • For IBM WebSphere Application Server 6.1.*x*, in addition to the `cacerts` certificate store, you must import the certificate into the following certificate store: *WEBSPHERE_HOME*/Web_Sphere/profiles/*SERVER_NAME*/config/cells/*CELL_NAME*/nodes/*NODE_NAME*/trust.p12 For example: C:/Web_Sphere/profiles/AppSrv01/config/cells/tcs055071Node01Cell/nodes/tcs055071Node0/trust.p12 • For IBM WebSphere Application Server 5.1.*x*, in addition to the `cacerts` certificate store, you must import the certificate into the following certificate store: *WEBSPHERE_HOME*/etc/DummyServerTrustFile.jks
JBoss Application Server	*JAVA_HOME*/jre/lib/security/cacerts

Table 4-1 (Cont.) Certificate Store Locations

Application Server	Certificate Store Location
Oracle Application Server	*ORACLE_HOME*/jdk/jre/lib/security/cacerts

4.2 Configuring Oracle Identity Governance

During application creation, if you did not choose to create a default form, then you must create a UI form for the application that you created by using the connector.

> **Note:**
>
> Perform the procedures described in this section only if you did not choose to create the default form during creating the application.

The following topics describe the procedures to configure Oracle Identity Governance:

- Creating and Activating a Sandbox
- Creating a New UI Form
- Publishing a Sandbox
- Updating an Existing Application Instance with a New Form

4.2.1 Creating and Activating a Sandbox

You must create and activate a sandbox to begin using the customization and form management features. You can then publish the sandbox to make the customizations available to other users.

See Creating a Sandbox and Activating a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

4.2.2 Creating a New UI Form

You can use Form Designer in Oracle Identity System Administration to create and manage application instance forms.

See Creating Forms By Using the Form Designer in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

While creating the UI form, ensure that you select the resource object corresponding to the newly created application that you want to associate the form with. In addition, select the **Generate Entitlement Forms** check box.

4.2.3 Publishing a Sandbox

Before publishing a sandbox, perform this procedure as a best practice to validate all sandbox changes made till this stage as it is difficult to revert the changes after a sandbox is published.

1. In Identity System Administration, deactivate the sandbox.

2. Log out of Identity System Administration.
3. Log in to Identity Self Service using the xelsysadm user credentials and then activate the sandbox that you deactivated in Step 1.
4. In the Catalog, ensure that the application instance form for your resource appears with correct fields.
5. Publish the sandbox. See Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

4.2.4 Updating an Existing Application Instance with a New Form

For any changes that you do in the schema of your application in Identity Self Service, you must create a new UI form and update the changes in an application instance.

To update an existing application instance with a new form:

1. Create and activate a sandbox.
2. Create a new UI form for the resource.
3. Open the existing application instance.
4. In the Form field, select the new UI form that you created.
5. Save the application instance.
6. Publish the sandbox.

See Also:

- Creating a Sandbox and Activating a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*
- Creating Forms By Using the Form Designer in *Oracle Fusion Middleware Administering Oracle Identity Governance*
- Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*

4.3 Harvesting Entitlements and Sync Catalog

You can populate Entitlement schema from child process form table, and harvest roles, application instances, and entitlements into catalog. You can also load catalog metadata.

To harvest entitlements and sync the catalog:

1. Run the scheduled jobs for lookup field synchronization listed in Reconciliation Jobs.
2. Run the Entitlement List scheduled job to populate Entitlement Assignment schema from the child process form table.
3. Run the Catalog Synchronization Job scheduled job.

> **See Also:**
>
> Predefined Scheduled Tasks in *Oracle Fusion Middleware Administering Oracle Identity Governance* for a description of the Entitlement List and Catalog Synchronization Job scheduled jobs

4.4 Managing Logging

Oracle Identity Governance uses the Oracle Diagnostic Logging (ODL) logging service for recording all types of events pertaining to the connector.

The following topics provide detailed information about logging:

- Understanding Log Levels
- Enabling logging

4.4.1 Understanding Log Levels

When you enable logging, Oracle Identity Governance automatically stores in a log file information about events that occur during the course of provisioning and reconciliation operations.

ODL is the principle logging service used by Oracle Identity Governance and is based on java.util.logger. To specify the type of event for which you want logging to take place, you can set the log level to one of the following:

- SEVERE.intValue()+100

  This level enables logging of information about fatal errors.

- SEVERE

  This level enables logging of information about errors that might allow Oracle Identity Governance to continue running.

- WARNING

  This level enables logging of information about potentially harmful situations.

- INFO

  This level enables logging of messages that highlight the progress of the application.

- CONFIG

  This level enables logging of information about fine-grained events that are useful for debugging.

- FINE, FINER, FINEST

  These levels enable logging of information about fine-grained events, where FINEST logs information about all events.

These message types are mapped to ODL message type and level combinations as shown in Table 4-2.

Table 4-2 Log Levels and ODL Message Type:Level Combinations

Java Level	ODL Message Type:Level
SEVERE.intValue()+100	INCIDENT_ERROR:1
SEVERE	ERROR:1
WARNING	WARNING:1
INFO	NOTIFICATION:1
CONFIG	NOTIFICATION:16
FINE	TRACE:1
FINER	TRACE:16
FINEST	TRACE:32

The configuration file for OJDL is logging.xml, which is located at the following path:

DOMAIN_HOME/config/fmwconfig/servers/*OIM_SERVER*/logging.xml

Here, *DOMAIN_HOME* and *OIM_SERVER* are the domain name and server name specified during the installation of Oracle Identity Governance.

4.4.2 Enabling logging

To enable logging in Oracle WebLogic Server:

1. Edit the logging.xml file as follows:
   a. Add the following blocks in the file:

```
<log_handler name="ebs-um-handler" level='[LOG_LEVEL]'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
<property name='logreader:' value='off'/>
     <property name='path' value='[FILE_NAME]'/>
     <property name='format' value='ODL-Text'/>
     <property name='useThreadName' value='true'/>
     <property name='locale' value='en'/>
     <property name='maxFileSize' value='5242880'/>
     <property name='maxLogSize' value='52428800'/>
     <property name='encoding' value='UTF-8'/>
   </log_handler>

<logger name='ORG.IDENTITYCONNECTORS.EBS' level='[LOG_LEVEL]'
useParentHandlers='false'>
     <handler name='ebs-um-handler'/>
     <handler name='console-handler'/>
   </logger>
```

   b. Replace both occurrences of **[LOG_LEVEL]** with the ODL message type and level combination that you require. Table 4-2 lists the supported message type and level combinations.

      Similarly, replace **[FILE_NAME]** with the full path and name of the log file in which you want log messages to be recorded.

      The following blocks show sample values for **[LOG_LEVEL]** and **[FILE_NAME]** :

```
<log_handler name='ebs-um-handler' level='NOTIFICATION:1'
class='oracle.core.ojdl.logging.ODLHandlerFactory'>
```

```
<property name='logreader:' value='off'/>
     <property name='path'
value='F:\MyMachine\middleware\user_projects\domains\base_domain1\servers
\oim_server1\logs\oim_server1-diagnostic-1.log'/>
     <property name='format' value='ODL-Text'/>
     <property name='useThreadName' value='true'/>
     <property name='locale' value='en'/>
     <property name='maxFileSize' value='5242880'/>
     <property name='maxLogSize' value='52428800'/>
     <property name='encoding' value='UTF-8'/>
   </log_handler>

<logger name='ORG.IDENTITYCONNECTORS.EBS' level='NOTIFICATION:1'
useParentHandlers='false'>
     <handler name='ebs-um-handler'/>
     <handler name='console-handler'/>
   </logger>
```

With these sample values, when you use Oracle Identity Governance, all messages generated for this connector that are of a log level equal to or higher than the `NOTIFICATION:1` level are recorded in the specified file.

2. Save and close the file.
3. Set the following environment variable to redirect the server logs to a file:

   For Microsoft Windows:

   ```
   set WLS_REDIRECT_LOG=FILENAME
   ```

   For UNIX:

   ```
   export WLS_REDIRECT_LOG=FILENAME
   ```

   Replace ***FILENAME*** with the location and name of the file to which you want to redirect the output.
4. Restart the application server.

4.5 Configuring the Connector for SSO

> **Note:**
>
> - Perform the procedure described in this section only if you want to configure the connector to work with a single sign-on solution during reconciliation and provisioning operations.
> - Before you perform this procedure, ensure that the connector for the LDAP-based repository of your single sign-on solution has been installed in your production environment.

You must perform the following steps to configure the connector for SSO:

1. Log in to the Design Console.
2. Modify the resource object as follows:
   a. Expand **Resource Management,** and then double-click **Resource Object.**

b. In the Name field, enter `Oracle EBS User Management` and then click **Search.**

c. On the Depends On tab, click **Assign.**

d. Select the resource object corresponding to your SSO target (for example, **OID User),** and then click **OK.**

e. Click the Save icon.

3. Modify the **Update SSO Attributes** process task to assign an event handler as follows:

a. Expand **Process Management,** and then double-click **Process Definition.**

b. Search for and open the **Oracle EBS UM User** process definition.

c. On the Tasks tab, double-click the **Update SSO Attributes** process task.

d. In the Editing Task: Update SSO Attributes dialog box, on the Integration tab, click **Add.**

The Handler Selection dialog box is displayed.

e. In the Handler Type region, select the **System** option, and then select the **CopyProcessFormData** event handler from the Handler Name region.

f. Click the Save icon.

g. In the confirmation dialog box that is displayed, click **OK.**

The CopyProcessFormData event handler is assigned to the process task.

4. Modify the **Create EBS User** process task to assign a generated task as follows:

a. On the Tasks tab of the Oracle EBS UM User process definition, double-click the **Create EBS User** process task.

The Editing Task: Create EBS User dialog box is displayed.

b. On the Responses tab, select the response code **SUCCESS.**

c. From the Tasks to Generate region, click **Assign.**

d. In the dialog box that is displayed, move the **Update SSO Attributes** task name from the right column to the left, and then click **OK.**

The Update SSO Attributes task is assigned to the process task.

e. Click the Save icon and close the Editing Task: Create EBS User dialog box.

5. Ensure that the lookup definition corresponding to the LDAP server that you are using exists and contains the right entries. For example, if you are using OID, then ensure the Lookup.Objects.OID User.Oracle EBS User Management.CopyAttributesMap exists and contains the following entry:

- **Code Key:** orclGuid
- **Decode:** SSO GUID

See Lookup Definitions Used During Connector Operations for a list of lookup definitions corresponding to your LDAP server.

6. Modify the Oracle EBS UM Application Instance as follows:

a. Log in to the System Administration console.

b. Create and activate a Sandbox. See Managing Sandboxes in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for instructions on creating and activating a sandbox

c. Modify the Oracle EBS UM Application Instance to specify the application instance of your SSO target (for example, OID) as a parent instance. See Modifying Application Instance Attributes in *Oracle Fusion Middleware Administering Oracle Identity Manager* for instructions on modifying an application instance.

d. Publish the sandbox. See Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for instructions on publishing a sandbox.

4.6 Localizing Field Labels in UI Forms

You can localize UI form field labels by using the resource bundle corresponding to the language you want to use. The resource bundles are available in the connector installation package.

To localize field label that is added to the UI forms:

1. Log in to Oracle Enterprise Governance.
2. In the left pane, expand **Application Deployments** and then select **oracle.iam.console.identity.sysadmin.ear.**
3. In the right pane, from the **Application Deployment** list, select **MDS Configuration.**
4. On the MDS Configuration page, click **Export** and save the archive to the local computer.
5. Extract the contents of the archive, and open the following file in a text editor:

   SAVED_LOCATION\xliffBundles\oracle\iam\ui\runtime\BizEditorBundle_en.xlf

   > **Note:**
   >
   > You will not be able to view the BizEditorBundle.xlf unless you complete creating the application for your target system or perform any customization such as creating a UDF.

6. Edit the BizEditorBundle.xlf file in the following manner:

   a. Search for the following text:

   ```
   <file source-language="en"
   original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
   datatype="x-oracle-adf">
   ```

   b. Replace with the following text:

   ```
   <file source-language="en" target-language="LANG_CODE"
   original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
   datatype="x-oracle-adf">
   ```

   In this text, replace *LANG_CODE* with the code of the language that you want to localize the form field labels. The following is a sample value for localizing the form field labels in Japanese:

   ```
   <file source-language="en" target-language="ja"
   original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
   datatype="x-oracle-adf">
   ```

c. Search for the application instance code. This procedure shows a sample edit for Oracle E-Business Suite application instance. The original code is:

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundl
e']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.
UD_EBS_UM_USRNAME__c_description']}">
<source>User Name</source>
<target/>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.EBSUMForm11.entity.EBSUMF
orm11EO.UD_EBS_UM_USRNAME__c_LABEL">
<source>User Name</source>
<target/>
</trans-unit>
```

d. Depending on the connector you are using, open the resource file (for example, EBS-UM.properties) from the connector package, and get the value of the attribute from the file, for example, global.udf.UD_EBS_UM_USER_NAME=\u4567d.

e. Replace the original code shown in Step 6.c with the following:

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundl
e']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.
UD_EBS_UM_USRNAME__c_description']}">
<source>User Name</source>
<target>\u4567d</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.EBSUMForm11.entity.EBSUMF
orm11EO.UD_EBS_UM_USRNAME__c_LABEL">
<source>User Name</source>
<target>\u4567d</target>
</trans-unit>
```

f. Repeat Steps 6.a through 6.d for all attributes of the process form.

g. Save the file as BizEditorBundle_*LANG_CODE*.xlf. In this file name, replace *LANG_CODE* with the code of the language to which you are localizing.

   Sample file name: BizEditorBundle_ja.xlf.

7. Repackage the ZIP file and import it into MDS.

> **See Also:**
>
> Deploying and Undeploying Customizations in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager,* for more information about exporting and importing metadata files

8. Log out of and log in to Oracle Identity Governance.

5

Using the EBS User Management Connector

You can use the EBS User Management connector for performing reconciliation and provisioning operations after configuring your application to meet your requirements.

5.1 Lookup Definitions Used During Connector Operations

Lookup definitions that are used during reconciliation and provisioning operations are either preconfigured or synchronized with the target system.

Lookup definitions used during connector operations can be categorized as follows:

5.1.1 Lookup Definitions Synchronized with the Target System

During a provisioning operation, you use a lookup field on the process form to specify a single value from a set of values. For example, you use the Responsibilities lookup field to select a responsibility to be assigned from the list of responsibilities in the lookup field. When you deploy the connector, lookup definitions corresponding to the lookup fields on the target system are created in Oracle Identity Governance. Lookup field synchronization involves copying additions or changes made to the target system lookup fields into the lookup definitions in Oracle Identity Governance.

The following is the format in which data is stored after lookup definition synchronization:

Code Key: <*IT_RESOURCE_KEY*>~<*LOOKUP_FIELD_VALUE*>

In this format:

- *IT_RESOURCE_KEY* is the numeric code assigned to each IT resource in Oracle Identity Governance.
- *LOOKUP_FIELD_VALUE* is the connector attribute value defined for code.

Sample value: `245~0`

Decode: <*IT_RESOURCE_NAME*>~<*LOOKUP_FIELD_VALUE*>

In this format:

- *IT_RESOURCE_KEY* is the name of the IT resource in Oracle Identity Governance.
- *LOOKUP_FIELD_VALUE* is the connector attribute value defined for decode.

Sample value: `Oracle EBS UM~FND`

During a provisioning operation, lookup fields are populated with values corresponding to the target system that you select for the operation.

5.1.2 Preconfigured Lookup Definitions for the EBS User Management Connector

This section discusses the other lookup definitions that are created in Oracle Identity Governance when you deploy the connector. These lookup definitions are either prepopulated with values or values must be manually entered in them after the connector is deployed. The other lookup definitions are as follows:

- Lookup.Oracle EBS UM.PartyType
- Lookup.Oracle EBS UM.PasswordExpTypes
- Lookup.Objects.EDIR User.Oracle EBS User Management.CopyAttributesMap
- Lookup.Objects.LDAP User.Oracle EBS User Management.CopyAttributesMap
- Lookup.Objects.OID User.Oracle EBS User Management.CopyAttributesMap
- Lookup.Objects.AD User.Oracle EBS User Management.CopyAttributesMap

5.1.2.1 Lookup.Oracle EBS UM.PartyType

The Lookup.Oracle EBS UM.PartyType lookup definition holds information about the types of parties that you can select for a target system account, which you create through Oracle Identity Governance.

The following is the format of the Code Key and Decode values in this lookup definition:

- **Code Key:** The type of party
- **Decode:** Description of the type of party

> **Note:**
>
> You cannot add new entries to this lookup definition.

Table 5-1 lists the default entries in this lookup definition.

Table 5-1 Entries in the Lookup.Oracle EBS UM.PartyType Lookup Definition

Code Key	Decode
Party	Party

Table 5-1 (Cont.) Entries in the Lookup.Oracle EBS UM.PartyType Lookup Definition

Code Key	Decode
Supplier	Supplier

5.1.2.2 Lookup.Oracle EBS UM.PasswordExpTypes

The Lookup.Oracle EBS UM.PasswordExpTypes lookup definition holds the options that you can select to specify when the password for the target system account (created through Oracle Identity Governance) must expire.

The following is the format of entries in this lookup definition:

- **Code Key:** The type of password expiry
- **Decode:** The type of password expiry

Table 5-2 lists the default entries in this lookup definition.

Table 5-2 Entries in the Lookup.Oracle EBS UM.PasswordExpTypes Lookup Definition

Code Key	Decode
Accesses	Accesses
Days	Days
None	None

5.1.2.3 Lookup.Objects.EDIR User.Oracle EBS User Management.CopyAttributesMap

The Lookup.Objects.EDIR User.Oracle EBS User Management.CopyAttributesMap lookup definition is used to configure the connector to work with an SSO solution during provisioning operations. In other words, this lookup definition is used when the target system is configured to use Oracle Access Governance to authenticate users. Oracle Access Governance in turn uses Novell eDirectory as an LDAP-based repository for storing user records.

The Lookup.Objects.EDIR User.Oracle EBS User Management.CopyAttributesMap lookup definition holds information that is used internally by an OIG adapter to copy field values from a Novell eDirectory account to the target system account. For example, the entries in the Lookup.Objects.EDIR User.Oracle EBS User Management.CopyAttributesMap lookup definition are used internally by the OIG adapter to copy the Reference ID value of a Novell eDirectory account to the SSO GUID field of the EBS UM account.

The following is the format of entries in this lookup definition:

- **Code Key:** Name of the field in the target system that must be populated with a value from a corresponding field in Novell eDirectory
- **Decode:** Corresponding field name in Novell eDirectory

Table 5-3 lists the default entries in this lookup definition.

Table 5-3 Entries in the Lookup.Objects.EDIR User.Oracle EBS User Management.CopyAttributesMap Lookup Definition

Code Key	Decode
Reference ID	SSO GUID

5.1.2.4 Lookup.Objects.LDAP User.Oracle EBS User Management.CopyAttributesMap

The Lookup.Objects.LDAP User.Oracle EBS User Management.CopyAttributesMap lookup definition is used when the target system is configured to use either Oracle Single Sign-On or Oracle Access Governance, to authenticate users. Oracle Single Sign-On and Oracle Access Governance in turn use an LDAP-based repository for storing user records.

The Lookup.Objects.LDAP User.Oracle EBS User Management.CopyAttributesMap lookup definition holds information is used internally by an OIG adapter to copy field values from an LDAP-based repository account to the target system account. For example, the entries in the Lookup.Objects.LDAP User.Oracle EBS User Management.CopyAttributesMap lookup definition are used internally by the OIG adapter to copy the NsuniqueID value of an LDAP account to the SSO GUID field of the EBS UM account.

The following is the format of entries in this lookup definition:

- **Code Key:** Name of the field in the target system that must be populated with a value from a corresponding field in any LDAP-based repository
- **Decode:** Corresponding field name in the LDAP-based repository

Table 5-4 lists the default entries in this lookup definition.

Table 5-4 Entries in the Lookup.Objects.LDAP User.Oracle EBS User Management.CopyAttributesMap Lookup Definition

Code Key	Decode
NsuniqueID	SSO GUID

5.1.2.5 Lookup.Objects.OID User.Oracle EBS User Management.CopyAttributesMap

The Lookup.Objects.OID User.Oracle EBS User Management.CopyAttributesMap lookup definition is used when the target system is configured to use Oracle Single Sign-On to authenticate users. Oracle Single Sign-On in turn uses Oracle Internet Directory as an LDAP-based repository for storing user records.

The Lookup.Objects.OID User.Oracle EBS User Management.CopyAttributesMap lookup definition holds information that is used internally by an OIG adapter to copy field values from an Oracle Internet Directory account to the target system account. For example, the entries in the Lookup.Objects.OID User.Oracle EBS User Management.CopyAttributesMap lookup definition are used internally by the OIG

adapter to copy the orclGuid value of an OID account to the SSO GUID field of the EBS UM account.

The following is the format of entries in this lookup definition:

- **Code Key:** Name of the field in the target system that must be populated with a value from a corresponding field in OID
- **Decode:** Corresponding field name in OID

Table 5-5 lists the default entries in this lookup definition.

Table 5-5 Entries in the Lookup.Objects.OID User.Oracle EBS User Management.CopyAttributesMap Lookup Definition

Code Key	Decode
orclGuid	SSO GUID

5.1.2.6 Lookup.Objects.AD User.Oracle EBS User Management.CopyAttributesMap

The Lookup.Objects.AD User.Oracle EBS User Management.CopyAttributesMap lookup definition is used when the target system is configured to use Oracle Single Sign-On to authenticate users. Oracle Single Sign-On in turn uses Active Directory as an LDAP-based repository for storing user records.

The Lookup.Objects.AD User.Oracle EBS User Management.CopyAttributesMap lookup definition holds information that is used internally by an OIG adapter to copy field values from a Microsoft Active Directory account to the target system account. For example, the entries in the Lookup.Objects.AD User.Oracle EBS User Management.CopyAttributesMap lookup definition are used internally by the OIG adapter to copy the Unique Id value of an AD account to the SSO GUID field of the EBS UM account.

The following is the format of entries in this lookup definition:

- **Code Key:** Name of the field in the target system that must be populated with a value from a corresponding field in AD
- **Decode:** Corresponding field name in AD

Table 5-5 lists the default entries in this lookup definition.

Table 5-6 Entries in the Lookup.Objects.AD User.Oracle EBS User Management.CopyAttributesMap Lookup Definition

Code Key	Decode
Unique Id	SSO GUID

5.2 About Reconciliation Queries and Provisioning Procedures

Reconciliation queries and provisioning procedures help the connector in performing reconciliation and provisioning operations efficiently.

5.2.1 About Reconciliation Queries

The User Management connector is configured to perform target resource reconciliation with the target system. Data from newly created and updated target system records is brought to Oracle Identity Governance and used to create and update Oracle E-Business Suite resources provisioned to OIG Users.

A SQL query is used to fetch target system records during reconciliation. All predefined SQL queries that are required to perform reconciliation are stored in the search.properties file. The search.properties file is a common file for all EBS Suite connectors. In other words, the search.properties file contains the queries for the EBS UM, HRMS Target, HRMS Trusted connectors.

When you run a scheduled job, the connector locates the corresponding SQL query in the search.properties file and then runs it on the target system database. Target system records that meet the query criteria are returned to Oracle Identity Governance.

Depending on your requirements, you can modify existing queries or add your own query in the search.properties. This is discussed later in this guide.

Information in the search.properties file is virtually divided into two parts. The first part lists entries containing the SQL query names in the following format:

OBJ_NAME.OP_NAME.MODE=QUERY_NAME

In this format:

- *OBJ_CLASS* is the name of the object class on which the reconciliation operation must be performed.
- *OP_NAME* is the type of reconciliation operation to be performed. A reconciliation operation can be a search op, sync op, or lookup op.
- *QUERY_NAME* is the name of the SQL query that is to be run on the target system database.

The second part lists the SQL query names and the corresponding SQL queries.

The following are the entries corresponding to the EBS UM connector in the search.properties file:

- __ACCOUNT__.search=UM_USER_RECON

  This query is used to reconcile all newly created and modified user records from the target system. The reconciliation operation that is performed is search based.
- __ACCOUNT__.sync=UM_USER_SYNC

This query is used to reconcile all newly created and modified user records from the target system. The reconciliation operation that is performed is sync based.

- __APPLICATIONS__.lookup=LOOKUP_APPLICATION_QUERY

  This query is used to synchronize values in the fnd_application table of the target system with the Lookup.Oracle EBS UM.Applications lookup definition in Oracle Identity Governance.

- __ROLES__.lookup=LOOKUP_ROLES_QUERY

  This query is used to synchronize values in the fnd_application table of the target system with the Lookup.Oracle EBS UM.Roles lookup definition in Oracle Identity Governance.

- __RESPONSIBILITIES__.lookup=LOOKUP_RESPONSIBILITY_QUERY

  This query is used to synchronize values in the fnd_responsibility_vl table of the target system with the Lookup.Oracle EBS UM.Responsibilities lookup definition in Oracle Identity Governance.

- __SECURITY_GROUPS__.lookup=LOOKUP_SECURITY_GROUP_QUERY

  This query is used to synchronize values in the fnd_security_groups table of the target system with the Lookup.Oracle EBS UM.SecurityGroups lookup definition in Oracle Identity Governance.

5.2.2 About Provisioning Procedures

Provisioning involves management of user accounts and assignment of responsibilities and roles to users in the target system. When you allocate (or provision) an Oracle E-Business Suite resource to an OIG User, the operation results in the creation of an account on Oracle E-Business Suite for that user. Similarly, when you update the resource on Oracle Identity Governance, the same update is made to the account on the target system.

The connector uses stored procedures for performing provisioning operations. The stored procedures are available in the wrapper packages of the target system.

Information about all stored procedures used for performing provisioning operations is defined in the Procedures.properties file. The same file contains stored procedures information for both the EBS UM and HRMS Target connectors.

When you perform a provisioning operation, the connector locates the corresponding stored procedure in the Procedures.properties file and then runs it on the target system to complete the provisioning operation.

Depending on your requirements, you can modify existing stored procedures or add your own stored procedures to the Procedures.properties file. This is discussed later in the guide.

The first property in the Procedures.properties file, DB.PACKAGES, lists all the wrapper packages that are used during connector operations. The subsequent entries in this file are in the following format:

OBJ_NAME.OP_NAME.TCA_TYPE=WRAPPER_PCKG.STORED_PROC

In this format:

- *OBJ_NAME* is the name of the object on which the provisioning operation must be performed.

- *OP_NAME* is the type of provisioning operation to be performed. For example, a provisioning operation can be either create, update, delete, enable, or disable.
- *TCA_TYPE* is the type of TCA record, whether party or supplier. *TCA_TYPE* is present only for entries corresponding to TCA record provisioning.
- *WRAPPER_PCKG* is the name of the wrapper package.
- *STORED_PROC* is the name of the stored procedure in the wrapper package that is to be run to on the target system to complete the provisioning operation.

The following are the entries corresponding to the EBS UM connector in the Procedures.properties file:

- **Entries corresponding to the __ACCOUNT__ object:**
  - __ACCOUNT__.create=OIM_FND_USER_TCA_PKG.CREATEUSER

    In this entry, the CREATEUSER stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Create User provisioning operation against the __ACCOUNT__ object.
  - __ACCOUNT__.create.userparty=OIM_FND_USER_TCA_PKG.CREATEUSERPARTY

    In this entry, the CREATEUSERPARTY stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for creating a user record with an existing TCA record.
  - __ACCOUNT__.validatepartyandperson=OIM_FND_USER_TCA_PKG.VALIDATEPARTYANDPERSON

    In this entry, the VALIDATEPARTYANDPERSON stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for validating person and party records before creating an account.
  - __ACCOUNT__.update=OIM_FND_USER_TCA_PKG.UPDATEUSER

    In this entry, the UPDATEUSER stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Update provisioning operation against the __ACCOUNT__ object.
  - __ACCOUNT__.enable=OIM_FND_USER_TCA_PKG.ENABLEUSER

    In this entry, the ENABLEUSER stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for enabling the user account of the __ACCOUNT__ object.
  - __ACCOUNT__.disable=OIM_FND_USER_TCA_PKG.DISABLEUSER

    In this entry, the DISABLEUSER stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for disabling the user account of the __ACCOUNT__ object.
  - __ACCOUNT__.update.username=OIM_FND_USER_TCA_PKG.CHANGE_USER_NAME

    In this entry, the CHANGE_USER_NAME stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Update user name provisioning operation against the __ACCOUNT__ object.
  - __ACCOUNT__.update.password=OIM_FND_USER_TCA_PKG.CHANGEPASSWORD

In this entry, the CHANGEPASSWORD stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Update user password provisioning operation against the __ACCOUNT__ object.

- __ACCOUNT__.update.userparty=OIM_FND_USER_TCA_PKG.UPDATEUSERPARTY

  In this entry, the UPDATEUSERPARTY stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Update user party provisioning operation against the __ACCOUNT__ object.

- __ACCOUNT__.delete=OIM_FND_USER_TCA_PKG.REVOKEUSER

  In this entry, the DELETE_PERSON_API stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Delete provisioning operation against the __ACCOUNT__ object.

- __ACCOUNT__.create.supplier=OIM_FND_USER_TCA_PKG.CREATE_SUPPLIER

  In this entry, the CREATE_SUPPLIER stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Create Supplier provisioning operation against the __ACCOUNT__ object.

- __ACCOUNT__.create.supplier_contact=OIM_FND_USER_TCA_PKG.CREATE_SUPPLIER_CONTACT

  In this entry, the CREATE_SUPPLIER_CONTACT stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Create Supplier Contact provisioning operation against the __ACCOUNT__ object.

- __ACCOUNT__.create.supplier_secattr=OIM_FND_USER_TCA_PKG.CREATE_SUPPLIER_SECURITY_ATTRS

  In this entry, the CREATE_SUPPLIER_SECURITY_ATTRS stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Create Security Attributes provisioning operation against the __ACCOUNT__ object.

- __ACCOUNT__.create.linkuser=OIM_FND_USER_TCA_PKG.LINK_USER_PARTY

  In this entry, the LINK_USER_PARTY stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for linking a user record with an existing party record. The LINK_USER_PARTY stored procedure is invoked soon after CREATEUSERPARTY stored procedure.

- __ACCOUNT__.create.party=OIM_FND_USER_TCA_PKG.CREATE_PARTY

  In this entry, the CREATE_PARTY stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for creating a new party record.

- __ACCOUNT__.update.party=OIM_FND_USER_TCA_PKG.UPDATE_PARTY

  In this entry, the UPDATE_PARTY stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for performing the Update Party record provisioning operation against the __ACCOUNT__ object.

- **Entries corresponding to child objects:**

- __RESPONSIBILITY__.add=OIM_FND_USER_TCA_PKG.ADDRESP

  In this entry, the ADDRESP stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for adding responsibilities for the __ACCOUNT__ object.

- __RESPONSIBILITY__.remove =OIM_FND_USER_TCA_PKG.DELRESP

  In this entry, the DELRESP stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for removing responsibilities for the __ACCOUNT__ object.

- __ROLE__.add=OIM_FND_USER_TCA_PKG.PROPAGATEUSERROLE

  In this entry, the PROPAGATEUSERROLE stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for adding roles for the __ACCOUNT__ object.

- __ROLE__.remove=OIM_FND_USER_TCA_PKG.REVOKEUSERROLE

  In this entry, the REVOKEUSERROLE stored procedure of the OIM_FND_USER_TCA_PKG wrapper package is used for removing roles for the __ACCOUNT__ object.

5.3 Configuring Reconciliation

You can configure the connector to specify the type of reconciliation and its schedule.

This section provides details on the following topics related to configuring reconciliation:

- Performing Full and Incremental Reconciliation
- Performing Limited Reconciliation
- Performing Batched Reconciliation

5.3.1 Performing Full and Incremental Reconciliation

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Governance. During incremental reconciliation, only records that are added or modified after the last reconciliation run are fetched into Oracle Identity Governance.

After you create the application, you must first perform full reconciliation. To perform a full reconciliation run, remove (delete) any value currently assigned to the Filter attribute of the Full User Reconciliation job. During a full reconciliation run, if you provide both batching parameters and filters, the connector processes the data in batches. Then, filters are applied to the processed data.

To perform incremental reconciliation, you can configure and run the Target Incremental User Reconciliation job.

See Reconciliation Jobs for information about these reconciliation jobs.

5.3.2 Performing Limited Reconciliation

You can perform limited reconciliation by creating filters for the reconciliation module, and reconcile records from the target system based on a specified filter criterion.

By default, all target system records that are added or modified after the last reconciliation run are reconciled during the current reconciliation run. You can customize this process by specifying the subset of added or modified target system records that must be reconciled.

This connector provides a Filter attribute (a scheduled job attribute) that allows you to use any of the Oracle EBS User Management resource attributes to filter the target system records.

When you specify a value for the Filter attribute, only the target system records that match the filter criterion are reconciled into Oracle Identity Governance. If you do not specify a value for the Filter attribute, then all the records in the target system are reconciled into Oracle Identity Governance.

You specify a value for the Filter attribute while configuring the user reconciliation scheduled job. The following are a few examples of the values for the Filter attribute:

- To reconcile all target system accounts whose user name is like 'jo*', use the filter `startsWith('user_name', 'jo')`.
- To reconcile all target system accounts whose email address is like '*@example.com', use the filter `endsWith('EMAIL_ADDRESS', '@example.com')`.
- To reconcile all target system accounts whose start date is later than 1st August, 2015, use the filter `greaterThan('START_DATE', 1438367400000)`. Note that the date value must be specified in milliseconds.

For detailed information about ICF Filters, see ICF Filter Syntax in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Governance*.

While creating the application, follow the instructions in Configuring Reconciliation Jobs to specify attribute values.

5.3.3 Performing Batched Reconciliation

You can perform batched reconciliation to reconcile a specific number of records from the target system into Oracle Identity Governance.

During a reconciliation run, all changes in the target system records are reconciled into Oracle Identity Governance. Depending on the number of records to be reconciled, this process may require a large amount of time. In addition, if the connection breaks during reconciliation, then the process would take longer to complete.

You can configure batched reconciliation to avoid these problems.

To configure batched reconciliation, you must specify value for the batchSize basic configuration parameter. Use this parameter to specify the number of records that must be included in each batch. By default, this value is set to `1000`.

You specify values for these attributes by following the instructions described in Configuring Reconciliation Jobs.

5.4 Configuring Reconciliation Jobs

Configure reconciliation jobs to perform reconciliation runs that check for new information on your target system periodically and replicates the data in Oracle Identity Governance.

You can apply this procedure to configure the reconciliation jobs for users and entitlements.

To configure a reconciliation job:

1. Log in to Identity System Administration.
2. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the scheduled job as follows:
   a. In the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
   b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the parameters of the scheduled task:
   - **Retries**: Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.
   - **Schedule Type**: Depending on the frequency at which you want the job to run, select the appropriate schedule type. See Creating Jobs in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

   In addition to modifying the job details, you can enable or disable a job.
5. On the **Job Details** tab, in the Parameters region, specify values for the attributes of the scheduled task.

> **Note:**
>
> Values (either default or user-defined) must be assigned to all the attributes. If even a single attribute value is left empty, then reconciliation is not performed.

6. Click **Apply** to save the changes.

> **Note:**
>
> You can use the Scheduler Status page in Identity System Administration to either start, stop, or reinitialize the scheduler.

5.5 Performing Provisioning Operations

You create a new user in Identity Self Service by using the Create User page. You provision or request for accounts on the Accounts tab of the User Details page.

To perform provisioning operations in Oracle Identity Governance:

1. Log in to Identity Self Service.
2. Create a user as follows:
   a. In Identity Self Service, click **Manage**. The Home tab displays the different Manage option. Click **Users**. The Manage Users page is displayed.
   b. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar. The Create User page is displayed with input fields for user profile attributes.
   c. Enter details of the user in the Create User page.
3. On the Account tab, click **Request Accounts**.
4. In the Catalog page, search for and add to cart the application instance for the connector that you configured earlier, and then click **Checkout**.
5. Specify value for fields in the application form and then click **Ready to Submit**.
6. Click **Submit**.

> **See Also:**
>
> Creating a User in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance* for details about the fields on the Create User page

5.6 Uninstalling the Connector

Uninstalling the connector deletes all the account-related data associated with its resource objects.

If you want to uninstall the connector for any reason, then run the Uninstall Connector utility. Before you run this utility, ensure that you set values for `ObjectType` and `ObjectValues` properties in the ConnectorUninstall.properties file. For example, if you want to delete resource objects, scheduled tasks, and scheduled jobs associated with the connector, then enter `"ResourceObject", "ScheduleTask", "ScheduleJob"` as the value of the `ObjectType` property and a semicolon-separated list of object values corresponding to your connector (for example, `GoogleApps User; GoogleApps Group`) as the value of the `ObjectValues` property.

> **Note:**
>
> If you set values for the `ConnectorName` and `Release` properties along with the `ObjectType`and `ObjectValue` properties, then the deletion of objects listed in the `ObjectValues` property is performed by the utility and the Connector information is skipped.

For more information, see Uninstalling Connectors in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

6

Extending the Functionality of the EBS User Management Connector

You can extend the functionality of the connector to address your specific business requirements.

- Adding New Multivalued Attributes for Reconciliation and Provisioning
- Configuring the Connector for Multiple Installations of the Target System
- Configuring Transformation and Validation of Data
- Configuring Action Scripts

6.1 Adding New Multivalued Attributes for Reconciliation and Provisioning

You can add new multivalued attributes for reconciliation and provisioning.

By default, the attributes listed in Attribute Mappings are mapped for reconciliation and provisioning between Oracle Identity Governance and the target system. If required, you can map additional multivalued attributes for reconciliation and provisioning. The following sections describe the procedures to be performed for adding new multivalued attributes. The **Security Attributes** multivalued attribute has been used as an example to illustrate these procedures.

- Summary of Steps to Add New Multivalued Attributes for Reconciliation and Provisioning
- Extending the Connector Schema
- Extending Oracle Identity Manager Metadata
- Creating Scheduled Jobs
- Updating the Connector Bundle
- Adding APIs to Wrapper Packages

6.1.1 Summary of Steps to Add New Multivalued Attributes for Reconciliation and Provisioning

The following a summary of high-level steps to be performed to add a new multivalued attribute for reconciliation and provisioning:

1. Update the DB wrapper package to include the new multivalued attribute. You must include the parent attribute in the main attribute list of the get_schema procedure and then create an attribute list with all the child attributes as described in Extending the Connector Schema.

2. Update Oracle Identity Goernance metadata to include the new attribute as described in Extending Oracle Identity Manager Metadata.
3. Create a scheduled job to synchronize values in the target system attributes corresponding to the newly created multivalued attribute with values in Oracle Identity Governance as described in Creating Scheduled Jobs.
4. Update the connector bundle to include the new multivalued attribute in the search.properties and Procedures.properties file as described in Updating the Connector Bundle.
5. Add APIs to Wrapper packages to enable provisioning operation on the newly added multivalued attribute as described in Adding APIs to Wrapper Packages.

6.1.2 Extending the Connector Schema

You must extend the connector schema to include a new multivalued attribute for reconciliation and provisioning. To do so:

1. Open any SQL client and connect to database using **APPS** user.
2. Open the body of the **OIM_FND_USER_TCA_PKG.pck** wrapper package.
3. Select the **get_schema()** stored procedure.
4. Declare the new multivalued attribute. The syntax for declaring the new multivalued attribute is as follows:

   ```
   attr := attributelist();
   ```

5. Initialize the attribute list by specifying the number of child attributes that the new multivalued attribute must contain in the following format:

   ```
   attr.extend(NUM);
   ```

   Here, *NUM* is the number of child attributes. Internally, an array for the specified number of child attributes is created.

   Sample value: `attr.extend(4);`

   You can also initialize the attribute list or increase the number of child attributes in the list by 1 by using the following statement for each child attribute to be added:

   ```
   attr.extend;
   ```

   > **See Also:**
   >
   > Sample Code Snippets for Extending the Connector Schema for sample code snippets

6. Define each child attribute to include information such as the attribute name, datatype, and permission flags in the following format:

   attr (*ORD_NO*) := attributeinfo(*ATTR_NAME*,*ATTR_TYPE*,*CREATE_FLAG*,*UPDATE_FLAG*,*REQUIRED_FLAG*,*READ_FLAG*)

   In this format:

   - *ORD_NO* is the order of the attribute in the list. This is mandatory.

- *ATTR_NAME* is the name of the child attribute.
- *ATTR_TYPE* is the SQL datatype of the child attribute.
- *CREATE_FLAG* is a flag to represent whether the attribute is required during a create provisioning operation.
- *UPDATE_FLAG* is a flag to represent whether the attribute can be updated.
- *REQUIRED_FLAG* is a flag to represent whether the attribute is mandatory.
- *READ_FLAG* is flag to represent whether the attribute can be read.

A value of `1` or `0` for each flag denotes True or False, respectively. For example, a value `1, 0, 1, 0` for the flags mean that the attribute is a mandatory attribute and must be considered during create provisioning operations.

7. End the new multivalued attribute definition and schema by using the following statements:

```
schemaout.extend;
schemaout(ORD_NO) := schema_object('ATTR_NAME', attr)
```

In this statement, *ORD_NO* is the order of the multivalued attribute in the connector schema and *ATTR_NAME* is the name of the multivalued attribute being added. The following are sample statements:

```
schemaout.extend;
schemaout( 4 ) := schema_object('__SECURITY_ATTRS__',attr);
```

8. Re-compile the wrapper package.

6.1.3 Extending Oracle Identity Manager Metadata

By default, the multivalued fields listed on the Schema page for your application in Identity Self Service are mapped for reconciliation between Oracle Identity Governance and the target system. If required, you can add new multivalued fields for target resource reconciliation.

To add new multivalued fields for reconciling users from a target application:

1. Log in to Oracle Identity System Administration and create a lookup that can hold the list of values for the multivalued field that you want to add.
2. Create a child form and add attributes as follows:
   a. Log in to Identity Self Service.
   b. Search for and open the application you created for your target system for editing.
   c. On the Schema page, add a new child form and its attributes. For example, enter values for the **Display Name** and Target Attribute fields.

> **Note:**
>
> - Ensure to select the **Recon Field** option.
> - When you add attributes to the child form, from the **Advanced Settings** option, ensure to mark the newly added attribute as a Lookup.
> - In the List of values field, enter the name of the lookup created in Step 1.

d. Apply the changes.

3. Log in to Identity System Administration, create a new form and associate it with your application.

6.1.4 Creating Scheduled Jobs

You can either use these predefined jobs or edit them to meet your requirements. Alternatively, you can create custom reconciliation jobs. For information about editing these predefined jobs or creating new ones, see Updating Reconciliation Jobs in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

6.1.5 Updating the Connector Bundle

You must update the connector bundle (org.identityconnectors.ebs-1.0.1115.jar) to include all the updates made in the earlier sections. To do so:

1. Download the connector bundle (org.identityconnectors.ebs-1.0.11150.jar) file from the Oracle Identity Governance database by running the Download JARs utility. This utility is copied into the following location when you install Oracle Identity Manager:

> **Note:**
>
> Before you use this utility, verify that the WL_HOME environment variable is set to the directory in which Oracle WebLogic Server is installed.

For Microsoft Windows:

OIM_HOME/server/bin/DownloadJars.bat

For UNIX:

OIM_HOME/server/bin/DownloadJars.sh

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded, and the location from which the JAR file is to be uploaded. Specify 4 as the value of the JAR type.

2. Extract the contents of the JAR file to any directory on the computer hosting Oracle Identity Governance.
3. In a text editor, open the search.properties file located in the configuration directory of the extracted JAR file.
4. In the first part of the search.properties file, add entries corresponding to the newly added attributes
5. In the first part of the search.properties file, add entries corresponding entries for the newly added attribute by defining the object name, type of reconciliation operation, and the SQL query name. For example, add the following entries:

   ```
   __SECURITY_ATTR_NAMES__.lookup=LOOKUP_SECATTR_NAME_QUERY
   __SECURITY_ATTR_TYPES__.lookup=LOOKUP_SECATTR_DATATYPE_QUERY
   ```

   In this example:

   - `__SECURITY_ATTR_NAMES` and `__SECURITY_ATTR_TYPES__` are the object names
   - `lookup` specifies that the query in this qntry will be used for performing lookup field synchronization.
   - `LOOKUP_SECATTR_NAME_QUERY` and `LOOKUP_SECATTR_DATATYPE_QUERY` are the SQL query names.
6. In the second part of the search.properties file, add the SQL query corresponding to the SQL query name specified in Step 5. For example, add the following entries:

   ```
   LOOKUP_SECATTR_DATATYPE_QUERY= select datatype as CODE, datatype as DECODE
   from ( select distinct(DATA_TYPE) as datatype from AK_ATTRIBUTES)

   LOOKUP_SECATTR_NAME_QUERY= select sa.ATTRIBUTE_CODE as CODE,
   (CONCAT(fa.application_short_name || '~', sa.ATTRIBUTE_CODE)) AS
   DECODE FROM fnd_application fa, AK_ATTRIBUTES sa where
   fa.application_id=sa.attribute_application_id
   ```
7. Update the SQL queries of UM_USER_RECON and UM_USER_SYNC to include information about the newly added attributes. For example, update both the UM_USER_RECON and UM_USER_SYNC SQL queries with the SQL query in Sample SQL Queries Updated to Include Multivalued Attributes.
8. Save and close the search.properties file.
9. In a text editor, open the Procedures.properties file located in the configuration directory of the JAR file extracted in Step 2.
10. Add entries to corresponding to the newly added attributes. For example, add the following entries:

    ```
    __SECURITY_ATTRS__.add=OIM_FND_USER_TCA_PKG.ADDUSERSECURITYATTRIBUTE
    __SECURITY_ATTRS__.remove=OIM_FND_USER_TCA_PKG.DELETEUSERSECURITYATTRIBUTE
    ```

    See About Provisioning Procedures for information about the format for adding entries to the Procedures.properties file.
11. Save and close the Procedures.properties file.
12. Re-create the connector bundle JAR file with the updated .properties files.
13. Run the Oracle Identity Governance Upload JARs utility to post the new connector bundle (updated in Step 12) to the Oracle Identity Governance database. This utility is copied into the following location when you install Oracle Identity Governance:

> **Note:**
>
> Before you use this utility, verify that the WL_HOME environment variable is set to the directory in which Oracle WebLogic Server is installed.

For Microsoft Windows:

OIM_HOME/server/bin/UploadJars.bat

For UNIX:

OIM_HOME/server/bin/UploadJars.sh

When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded, and the location from which the JAR file is to be uploaded. Specify 4 as the value of the JAR type.

6.1.6 Adding APIs to Wrapper Packages

You must add APIs to Wrappers packages to enable the connector to perform provisioning operations on the newly added attribute. To do so:

1. Open any SQL client. For example, SQL Developer.
2. Open specification of the **OIM_FND_USER_TCA_PKG** package and then add entries that define the methods and their input parameters for performing provisioning operations. For example, add the following methods for the newly added attribute:

```
procedure addUserSecurityAttribute(user_id in number, SECURITY_ATTR_NAME
in varchar2, SECURITY_APP_ID varchar2,SECURITY_ATTR_VALUE
varchar2,SECURITY_ATTR_TYPE varchar2);
procedure deleteUserSecurityAttribute(user_id in number, SECURITY_ATTR_NAME
in varchar2, SECURITY_APP_ID varchar2,SECURITY_ATTR_VALUE
varchar2,SECURITY_ATTR_TYPE varchar2);
```

3. Open the **OIM_FND_USER_TCA_PKG** package body and add the implementation of methods defined in the preceding step. For example, add the following implementation for the newly added attribute:

```
procedure addUserSecurityAttribute(user_id in number, SECURITY_ATTR_NAME
in varchar2, SECURITY_APP_ID varchar2,SECURITY_ATTR_VALUE
varchar2,SECURITY_ATTR_TYPE varchar2)
    IS
          x_return_status VARCHAR2(2000);
          x_msg_count NUMBER;
          x_msg_data VARCHAR2(2000);
  l_varchar2_value varchar2(2000);
  l_date_value date;
  l_number_value NUMBER;
    begin
    if SECURITY_ATTR_TYPE = 'NUMBER' then
l_number_value := SECURITY_ATTR_VALUE;
elsif SECURITY_ATTR_TYPE = 'DATE' then
    l_date_value := SECURITY_ATTR_VALUE;
else
```

```
l_varchar2_value := SECURITY_ATTR_VALUE;
end if;
        icx_user_sec_attr_pub.create_user_sec_attr(
           p_api_version_number   => 1,
           p_return_status        => x_return_status,
           p_msg_count            => x_msg_count,
           p_msg_data             => x_msg_data,
           p_web_user_id          => user_id,
           p_attribute_code       => SECURITY_ATTR_NAME,
           p_attribute_appl_id    => SECURITY_APP_ID,
           p_varchar2_value       => l_varchar2_value,
           p_date_value           => l_date_value,
           p_number_value         => l_number_value,
           p_created_by           => -1,
           p_creation_date        => sysdate,
           p_last_updated_by      => -1,
           p_last_update_date     => sysdate,
           p_last_update_login    => -1);
    end addUserSecurityAttribute;

procedure deleteUserSecurityAttribute(user_id in number, SECURITY_ATTR_NAME
in varchar2, SECURITY_APP_ID varchar2,SECURITY_ATTR_VALUE
varchar2,SECURITY_ATTR_TYPE varchar2)
    IS
          x_return_status VARCHAR2(2000);
          x_msg_count NUMBER;
          x_msg_data VARCHAR2(2000);
  l_varchar2_value varchar2(2000);
  l_date_value date;
  l_number_value NUMBER;
    begin
    if SECURITY_ATTR_TYPE = 'NUMBER' then
l_number_value := SECURITY_ATTR_VALUE;
elsif SECURITY_ATTR_TYPE = 'DATE' then
    l_date_value := SECURITY_ATTR_VALUE;
else
l_varchar2_value := SECURITY_ATTR_VALUE;
end if;
        icx_user_sec_attr_pub.Delete_User_Sec_Attr(
           p_api_version_number   => 1,
           p_return_status        => x_return_status,
           p_msg_count            => x_msg_count,
           p_msg_data             => x_msg_data,
           p_web_user_id          => user_id,
           p_attribute_code       => SECURITY_ATTR_NAME,
           p_attribute_appl_id    => SECURITY_APP_ID,
           p_varchar2_value       => l_varchar2_value,
           p_date_value           => l_date_value,
           p_number_value         => l_number_value
    );
    end deleteUserSecurityAttribute;
```

4. Save and close the file.
5. Rerun the scripts to compile the wrapper package.

6.2 Configuring the Connector for Multiple Installations of the Target System

You must create copies of configurations of your base application to configure it for multiple installations of the target system.

The following example illustrates this requirement:
The London and New York offices of Example Multinational Inc. have their own installations of the target system, including independent schema for each. The company has recently installed Oracle Identity Governance, and they want to configure it to link all the installations of the target system.

To meet the requirement posed by such a scenario, you must clone your application which copies all configurations of the base application into the cloned application. For more information about cloning applications, see Cloning Applications in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

6.3 Configuring Transformation and Validation of Data

Configure transformation and validation of user account data by writing Groovy script logic while creating your application.

You can configure transformation of reconciled single-valued user data according to your requirements. For example, you can use First Name and Last Name values to create a value for the Full Name field in Oracle Identity Governance.

Similarly, you can configure validation of reconciled and provisioned single-valued data according to your requirements. For example, you can validate data fetched from the First Name attribute to ensure that it does not contain the number sign (#). In addition, you can validate data entered in the First Name field on the process form so that the number sign (#) is not sent to the target system during provisioning operations.

To configure transformation or validation of user account data, you must write Groovy scripts while creating your application. For more information about writing Groovy script-based validation and transformation logic, see Validation and Transformation of Provisioning and Reconciliation Attributes of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

6.4 Configuring Action Scripts

You can configure **Action Scripts** by writing your own Groovy scripts while creating your application.

These scripts can be configured to run before or after the create, update, or delete an account provisioning operations. For example, you can configure a script to run before every user creation operation.

For information on adding or editing action scripts, see Updating the Provisioning Configuration in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance*.

7

Upgrading the EBS User Management Connector

If you have already deployed the 11.1.1.5.0 version of this connector, then upgrade the connector to the current release 12.2.1.3.0.

> **Note:**
>
> Before you perform the upgrade procedure:
>
> - It is strongly recommended that you create a backup of the Oracle Identity Manager database. Refer to the database documentation for information about creating a backup.
> - As a best practice, perform the upgrade procedure in a test environment initially.

- Preupgrade Steps
- Upgrade Steps
- Postupgrade Steps

> **See Also:**
>
> Upgrading Connectors in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information on these steps

7.1 Preupgrade Steps

Preupgrade steps for the connector involves performing a reconciliation run to fetch records from the target system, defining the source connector in Oracle Identity Manager, creating copies of the connector if you want to configure it for multiple installations of the target system, and disabling all the scheduled jobs.

Perform the following preupgrade steps:

1. Perform a reconciliation run to fetch all latest updates to Oracle Identity Manager.
2. Perform the preupgrade procedure documented in Managing Connector Lifecycle in *Oracle Fusion Middleware Administering Oracle Identity Manager*.
3. Define the source connector (an earlier release of the connector that must be upgraded) in Oracle Identity Manager. You define the source connector to update the Deployment Manager XML file with all customization changes made

to the connector. See Managing Connector Lifecycle in *Oracle Fusion Middleware Administering Oracle Identity Manager* for more information.

4. If required, create the connector XML file for a clone of the source connector.
5. Disable all the scheduled jobs.

7.2 Upgrade Steps

This is a summary of the procedure to upgrade the connector for both staging and production environments.

Depending on the environment in which you are upgrading the connector, perform one of the following steps:

- Staging Environment

  Perform the upgrade procedure by using the wizard mode.

  > **Note:**
  >
  > Do not upgrade IT resource type definition. In order to retain the default setting, you must map the IT resource definition to "None".

- Production Environment

  Perform the upgrade procedure by using the silent mode.

See Managing Connector Lifecycle in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about the wizard and silent modes.

7.3 Postupgrade Steps

Depending on the connector you are upgrading from, perform one of the following procedures::

- Postupgrade Steps for the Oracle EBS UM TCA Connector from Release 9.1.0.7.*x* to 11.*x*
- Postupgrade Steps for the Oracle EBS UM TCA Connector from Release 11.*x* to this Release
- Postupgrade Steps for the Oracle EBS UM Connector from Release 9.1.0.7.*x* to 11.*x*
- Postupgrade Steps for the Oracle EBS UM Connector from Release 11.*x* to this Release

7.3.1 Postupgrade Steps for the Oracle EBS UM TCA Connector from Release 9.1.0.7.*x* to 11.*x*

Perform the following procedure if you are upgrading the Oracle EBS UM TCA connector from release 9.1.0.7.*x* to this release:

1. Download the latest version of this connector from Oracle Technology Network and extract its contents to any directory on the computer hosting Oracle Identity Manager.
2. Run the Upload JARs utility to post the latest version of the connector bundle JAR file (org.identityconnectors.ebs-1.0.1115.jar) from the /bundle directory of the installation media to the Oracle Identity Governance database.

   For Microsoft Windows:

   OIM_HOME/server/bin/UploadJars.bat

   For UNIX:

   OIM_HOME/server/bin/UploadJars.sh

   When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded (specify the JAR type as ICFBundle, option 4), and the location from which the JAR file is to be uploaded.
3. Run the Form Version Control (FVC) utility to manage data changes on a form after an upgrade operation. To do so:
   a. In a text editor, open the fvc.properties file located in the *OIM_DC_HOME* directory and include the following entries:

      ```
      ResourceObject;Oracle EBS User Management
      FormName;UD_EBST_USR
      FromVersion;SPECIFY_THE_VERSION_OF_FORM_THAT_WAS_IN_THE_ACTIVE_STATUS_BEFORE_THE_UPGRADE
      ToVersion;SPECIFY_THE_VERSION_OF_FORM_THAT_IS_IN_THE_ACTIVE_STATUS_AFTER_THE_UPGRADE
      ```

   b. Run the FVC utility. This utility is copied into the following directory when you install the design console:

      For Microsoft Windows:

      OIM_DC_HOME/fvcutil.bat

      For UNIX:

      OIM_DC_HOME/fvcutil.sh

      When you run this utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, and the logger level and log file location.
4. Run the Post Upgrade Script as follows:
   a. Connect to the Oracle Identity Manager database by using the OIG User credentials.
   b. Run the **PostUpgradeScript_TCAEBSUM.sql** script located in the *OIM_HOME*/server/ConnectorDefaultDirectory/EBSUM_PCKG/upgrade directory.
5. Configure the upgraded IT resource of the source connector.
6. Change the literal value for child forms as follows:
   a. Log in to the Design Console.
   b. Expand **Process Management,** and then double-click **Process Definition.**

c. Search for and open the **Oracle EBS UM User** process definition.

d. On the Tasks tab, double-click the **Add User Responsibility** process task.

   The Editing Task: Add User Responsibility dialog box is displayed.

e. On the Integration tab, double-click the **childTableName** adapter variable.

   The Edit Mapping for Variable dialog box is displayed.

f. In the Literal Value field, change the value from `UD_UM_RESP` to `UD_EBST_RSP`.

g. Click the Save icon and close the dialog box.

h. Repeat Steps 6.d through 6.g for the **Update User Responsibility** and **Remove User Responsibility** process tasks.

i. Repeat Steps 6.d through 6.g for the Add User Role, Update User Role, and Remove User Role process tasks by changing the value of the Literal Value field from `UD_UM_ROLE` to `UD_EBST_RLS`.

7. Change the name of the child form in the **Lookup.Oracle EBS UM.UM.ProvAttrMap** lookup definition as follows:

   a. Expand **Administration,** and then double-click **Lookup Definition.**

   b. Search for and open the **Lookup.Oracle EBS UM.UM.ProvAttrMap** lookup definition.

   c. Search for all entries beginning with **UD_UM_RESP** and replace it with `UD_EBST_RSP`. For example, replace the UD_UM_RESP~Application Name[LOOKUP] entry with `UD_EBST_RSP~Application Name[LOOKUP]`.

      Similarly, search for all entries beginning with **UD_UM_ROLE** and replace it `with UD_EBST_RLS`. For example, replace the UD_UM_ROLE~Role Start Date[DATE] entry with `UD_EBST_RLS~Role Start Date[DATE]`.

   d. Click the Save icon.

8. Change the literal value for the parent form as follows:

   a. Expand **Development Tools,** and then double-click **Form Designer.**

   b. Search for and open the **UD_EBST_USR** form.

   c. On the Additional Columns tab, double-click the **UD_EBS_UM Updated** process form.

      The Editing Task: UD_EBS_UM Updated dialog box is displayed.

   d. On the Integration tab, change the literal value from **UD_EBS_UM** to `UD_EBST_USR`.

   e. Click the Save icon and close the dialog box.

9. Remove the old prepopulate adapter associated with the process form field as follows:

   a. Expand **Development Tools,** and then double-click **Form Designer.**

   b. Search for and open the **UD_EBST_USR** form.

   c. Create a new version (for example, **v_11.1.1.5.0_1**) of the form and save it.

   d. Select the newly created form version.

   e. On the Pre-Populate tab, select the row containing the old prepopulate adapter **EBSPrePopFirstName**, and then click **Delete**.

f. Click **OK** in the Alert dialog box to confirm that you want to proceed with deleting the prepopulate adapter.

g. Repeat Steps 9.e and 9.f to delete the **EBSPrePopLastName** prepopulate adapter associated with the Party Last Name form field.

h. Click the Save icon and then Click **Make Version Active**.

10. Update the localization properties. To do so, you must update the resource bundle of a user locale with new names of the process form attributes for proper translations after upgrading the connector. You can modify the properties file of your locale in the resources directory of the connector bundle.

    For example, the process form (UD_EBS_UM) attributes are referenced in the Japanese properties file, EBS-UM_ja.properties, as global.udf.UD_EBS_UM_PARTY_FNAME. During upgrade, the process form name is changed to old form name UD_EBST_USR (in case of EBS UM TCA upgrade) or UD_EBS_USER (in case of EBS Plain UM upgrade) to global.udf.UD_EBS_UM_PARTY_FNAME. Therefore, you must add the process form attributes to global.udf.UD_EBS_UM_PARTY_FNAME.

11. Restart Oracle Identity Manager. Alternatively, you can purge the cache for the changes to reflect in Oracle Identity Governance. See Purging Cache in *Oracle Fusion Middleware Administering Oracle Identity Manager* for more information about the PurgeCache utility.

12. Replicate all the changes made to the Form Designer of the Design Console to a new UI form as follows:

    a. Log in to Oracle Identity System Administration.

    b. Create and active a sandbox. See Creating and Activating a Sandbox for more information.

    c. Create a new UI form to view the upgraded fields. See Creating a New UI Form for more information about creating a UI form.

    d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in the previous step), and then save the application instance.

    e. Publish the sandbox. See Publishing a Sandbox for more information.

After upgrading the connector, you can perform either full reconciliation or incremental reconciliation. This ensures that records created or modified since the last reconciliation run (the one that you performed in Preupgrade Steps) are fetched into Oracle Identity Manager. From the next reconciliation run onward, the reconciliation engine automatically enters a value for the Latest Token attribute.

Before you perform lookup field synchronization, ensure to remove all preupgrade entries from the lookup definitions Oracle Identity Governance. After upgrade these values must be synchronized with the lookup fields in the target system.

See Performing Full and Incremental Reconciliation for more information about performing full or incremental reconciliation.

7.3.2 Postupgrade Steps for the Oracle EBS UM TCA Connector from Release 11.*x* to this Release

Perform the following procedure if you are upgrading the Oracle EBS UM TCA connector from release 9.1.0.7.*x* to this release.

1. Download the latest version of this connector from Oracle Technology Network and extract its contents to any directory on the computer hosting Oracle Identity Manager.
2. Run the Upload JARs utility to post the latest version of the connector bundle JAR file (org.identityconnectors.ebs-1.0.1115.jar) from the /bundle directory of the installation media to the Oracle Identity Governance database.

   For Microsoft Windows:

   OIM_HOME/server/bin/UploadJars.bat

   For UNIX:

   OIM_HOME/server/bin/UploadJars.sh

   When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded (specify the JAR type as ICFBundle, option 4), and the location from which the JAR file is to be uploaded.
3. Run the Form Version Control (FVC) utility to manage data changes on a form after an upgrade operation. To do so:
   a. In a text editor, open the fvc.properties file located in the *OIM_DC_HOME* directory and include the following entries:

      ```
      ResourceObject;Oracle EBS User Management
      FormName;UD_EBST_USR
      FromVersion;SPECIFY_THE_VERSION_OF_FORM_THAT_WAS_IN_THE_ACTIVE_STATUS_BEFORE_THE_UPGRADE
      ToVersion;SPECIFY_THE_VERSION_OF_FORM_THAT_IS_IN_THE_ACTIVE_STATUS_AFTER_THE_UPGRADE
      ```

   b. Run the FVC utility. This utility is copied into the following directory when you install the design console:

      For Microsoft Windows:

      OIM_DC_HOME/fvcutil.bat

      For UNIX:

      OIM_DC_HOME/fvcutil.sh

      When you run this utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, and the logger level and log file location.
4. Run the Post Upgrade Script as follows:
   a. Connect to the Oracle Identity Manager database by using the OIG User credentials.

b. Run the **PostUpgradeScript_TCAEBSUM.sql** script located in the *OIM_HOME*/server/ConnectorDefaultDirectory/EBSUM_PCKG/upgrade directory.

7.3.3 Postupgrade Steps for the Oracle EBS UM Connector from Release 9.1.0.7.*x* to 11.*x*

Perform the following procedure if you are upgrading the Oracle EBS UM connector from release 9.1.0.7.*x* to 11.*x*:

1. Download the latest version of this connector from Oracle Technology Network and extract its contents to any directory on the computer hosting Oracle Identity Governance.
2. Run the Upload JARs utility to post the latest version of the connector bundle JAR file (org.identityconnectors.ebs-1.0.1115.jar) from the /bundle directory of the installation media to the Oracle Identity Governance database.

   For Microsoft Windows:

   OIM_HOME/server/bin/UploadJars.bat

   For UNIX:

   OIM_HOME/server/bin/UploadJars.sh

   When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded (specify the JAR type as ICFBundle, option 4), and the location from which the JAR file is to be uploaded.
3. Run the Form Version Control (FVC) utility to manage data changes on a form after an upgrade operation. To do so:

   a. In a text editor, open the fvc.properties file located in the *OIM_DC_HOME* directory and include the following entries:

   ```
   ResourceObject;Oracle EBS User Management
   FormName;UD_EBS_USER
   FromVersion;SPECIFY_THE_VERSION_OF_FORM_THAT_WAS_IN_THE_ACTIVE_STATUS_BEFORE_THE_UPGRADE
   ToVersion;SPECIFY_THE_VERSION_OF_FORM_THAT_IS_IN_THE_ACTIVE_STATUS_AFTER_THE_UPGRADE
   ```

   b. Run the FVC utility. This utility is copied into the following directory when you install the design console:

   For Microsoft Windows:

   OIM_DC_HOME/fvcutil.bat

   For UNIX:

   OIM_DC_HOME/fvcutil.sh

   When you run this utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, and the logger level and log file location.
4. Run the Post Upgrade Script as follows:

a. Connect to the Oracle Identity Manager database by using the OIG User credentials.

b. Run the **PostUpgradeScript_PlainEBSUM.sql** script located in the *OIM_HOME*/server/ConnectorDefaultDirectory/EBSUM_PCKG/upgrade directory.

5. Configure the upgraded IT resource of the source connector.

6. Change the literal value for child forms as follows:

a. Log in to the Design Console.

b. Expand **Process Management,** and then double-click **Process Definition.**

c. Search for and open the **Oracle EBS UM User** process definition.

d. On the Tasks tab, double-click the **Add User Responsibility** process task.

The Editing Task: Add User Responsibility dialog box is displayed.

e. On the Integration tab, double-click the **childTableName** adapter variable.

The Edit Mapping for Variable dialog box is displayed.

f. In the Literal Value field, change the value from `UD_UM_RESP` to `UD_EBS_RESP`.

g. Click the Save icon and close the dialog box.

h. Repeat Steps 6.d through 6.g for the **Update User Responsibility** and **Remove User Responsibility** process tasks.

i. Repeat Steps 6.d through 6.g for the Add User Role, Update User Role, Remove User Role process tasks by changing the value of the Literal Value field from `UD_UM_ROLE` to `UD_EBS_RLS`.

7. Change the name of the child form in the **Lookup.Oracle EBS UM.UM.ProvAttrMap** lookup definition as follows:

a. Expand **Administration,** and then double-click **Lookup Definition.**

b. Search for and open the **Lookup.Oracle EBS UM.UM.ProvAttrMap** lookup definition.

c. In the Code Key column, search for all entries beginning with **UD_UM_RESP** and replace it with `UD_EBS_RESP`. For example, replace the UD_UM_RESP~Application Name[LOOKUP] entry with `UD_EBS_RESP~Application Name[LOOKUP]`.

Similarly, search for all entries beginning with **UD_UM_ROLE** and replace it `with UD_EBS_RLS`. For example, replace the UD_UM_ROLE~Role Start Date[DATE] entry with `UD_EBS_RLS~Role Start Date[DATE]`.

d. Click the Save icon.

8. Change the literal value for the parent form as follows:

a. Expand **Development Tools,** and then double-click **Form Designer.**

b. Search for and open the **UD_EBST_USR** form.

c. On the Additional Columns tab, double-click the **UD_EBS_UM Updated** process form.

The Editing Task: UD_EBS_UM Updated dialog box is displayed.

d. On the Integration tab, change the literal value from **UD_EBS_UM** to `UD_EBS_USER`.

e. Click the Save icon and close the dialog box.

9. Remove the old repopulate adapter associated with the process form field as follows:

   a. Expand **Development Tools** and then double-click **Form Designer.**

   b. Search for and open the **UD_EBS_USER** form.

   c. Create a new version (for example, **v_11.1.1.5.0_1**) of the form and save it.

   d. Select the newly created form version.

   e. On the Pr-Populate tab, select the row containing the old repopulate adapter **Disproportionate**, and then click **Delete**.

   f. Click **OK** in the Alert dialog box to confirm that you want to proceed with deleting the repopulate adapter.

   g. Click the Save icon and then click **Make Version Active**.

10. Update the localization properties. To do so, you must update the resource bundle of a user locale with new names of the process form attributes for proper translations after upgrading the connector. You can modify the properties file of your locale in the resources directory of the connector bundle.

    For example, the process form (UD_EBS_UM) attributes are referenced in the Japanese properties file, ENS-UM_ja.properties, as global.udf.UD_EBS_UM_PARTY_FNAME. During upgrade, the process form name is changed to old form name UD_EBST_USR (in case of EBS UM TCA upgrade) or UD_EBS_USER (in case of EBS Plain UM upgrade) to global.udf.UD_EBS_UM_PARTY_FNAME. Therefore, you must add the process form attributes to global.udf.UD_EBS_UM_PARTY_FNAME.

11. Restart Oracle Identity Governance. Alternatively, you can purge the cache for the changes to reflect in Oracle Identity Manager. See Purging Cache in *Oracle Fusion Middleware Administering Oracle Identity Manager* for more information about the Purge Cache utility.

12. Replicate all the changes made to the Form Designer of the Design Console to a new UI form as follows:

    a. Log in to Oracle Identity System Administration.

    b. Create and active a sandbox. See Creating and Activating a Sandbox for more information.

    c. Create a new UI form to view the upgraded fields. See Creating a New UI Form for more information about creating a UI form.

    d. Associate the newly created UI form with the application instance of your target system. To do so, open the existing application instance for your resource, from the Form field, select the form (created in the previous step), and then save the application instance.

    e. Publish the sandbox. See Publishing a Sandbox for more information.

After upgrading the connector, you can perform either full reconciliation or incremental reconciliation. This ensures that records created or modified since the last reconciliation run (the one that you performed in Preupgrade Steps) are fetched into Oracle Identity Governance. From the next reconciliation run onward, the reconciliation engine automatically enters a value for the Latest Token attribute.

Before you perform lookup field synchronization, ensure to remove all upgradeable entries from the lookup definitions Oracle Identity Manager. After upgrade these values must be synchronized with the lookup fields in the target system.

See Performing Full and Incremental Reconciliation for more information about performing full or incremental reconciliation.

7.3.4 Postupgrade Steps for the Oracle EBS UM Connector from Release 11.*x* to this Release

Perform the following procedure if you are upgrading the Oracle EBS UM connector from release 9.1.0.7.*x* to this release:

1. Download the latest version of this connector from Oracle Technology Network and extract its contents to any directory on the computer hosting Oracle Identity Governance.
2. Run the Upload JARs utility to post the latest version of the connector bundle JAR file (org.identityconnectors.ebs-1.0.1115.jar) from the /bundle directory of the installation media to the Oracle Identity Governance database.

   For Microsoft Windows:

   OIM_HOME/server/bin/UploadJars.bat

   For UNIX:

   OIM_HOME/server/bin/UploadJars.sh

   When you run the utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, URL of the Oracle Identity Governance host computer, context factory value, type of JAR file being uploaded (specify the JAR type as ICFBundle, option 4), and the location from which the JAR file is to be uploaded.
3. Run the Form Version Control (FVC) utility to manage data changes on a form after an upgrade operation. To do so:
   a. In a text editor, open the fvc.properties file located in the *OIM_DC_HOME* directory and include the following entries:

      ```
      ResourceObject;Oracle EBS User Management
      FormName;UD_EBS_USER
      FromVersion;SPECIFY_THE_VERSION_OF_FORM_THAT_WAS_IN_THE_ACTIVE_STATUS_BEFORE_THE_UPGRADE
      ToVersion;SPECIFY_THE_VERSION_OF_FORM_THAT_IS_IN_THE_ACTIVE_STATUS_AFTER_THE_UPGRADE
      ```

   b. Run the FVC utility. This utility is copied into the following directory when you install the design console:

      For Microsoft Windows:

      OIM_DC_HOME/fvcutil.bat

      For UNIX:

      OIM_DC_HOME/fvcutil.sh

      When you run this utility, you are prompted to enter the login credentials of the Oracle Identity Governance administrator, and the logger level and log file location.

4. Run the Post Upgrade Script as follows:
    a. Connect to the Oracle Identity Manager database by using the OIG User credentials.
    b. Run the **PostUpgradeScript_PlainEBSUM.sql** script located in the *OIM_HOME*/server/ConnectorDefaultDirectory/EBSUM_PCKG/upgrade directory.

A

Sample SQL Queries for the UM_USER_RECON and UM_USER_SYNC SQL Query Names

This appendix lists sample SQL queries that can be used to update the UM_USER_RECON and UM_USER_SYNC queries in the search.properties file. This appendix contains the following sections:

A.1 Sample SQL Queries Updated to Include Single-Valued Attributes

Use this SQL query to update the UM_USER_RECON and UM_USER_SYNC queries.

If you have added a single-valued attribute (Customer Id) as part of performing the procedure described in Attribute Mappings:

```
with roledata as ( \
               select
fa.application_id,fa.application_short_name,wlr.name,ura.user_name
user_name ,ura.start_date active_from,ura.end_date active_to from wf_local_roles
wlr,wf_user_role_assignments ura,fnd_application fa where ura.role_name like
'UMX%' AND wlr.parent_orig_system = 'UMX' and  wlr.name=ura.role_name and
fa.application_short_name = wlr.owner_tag and  ( (ura.start_date <
nvl(ura.end_date, TO_DATE('31-DEC-4712','dd-mon-yyyy')) and ura.start_date >
sysdate) or sysdate between ura.start_date and nvl(ura.end_date, TO_DATE('31-
DEC-4712','dd-mon-yyyy')) ) \
               ) , party as ( \
               select USER_ID AS user_id,USER_GUID AS user_guid,sysdate  as
system_date, LAST_UPDATE_DATE DATE_UPDATED, case when password_lifespan_days > 0
then 'Days' when password_lifespan_accesses > 0 then 'Accesses' else 'None' end
as PASSWORD_EXP_TYPE, case when password_lifespan_days > 0 then
password_lifespan_days when password_lifespan_accesses > 0 then
password_lifespan_accesses else null end as PASSWORD_LIFESPAN, EMPLOYEE_ID as
EMPLOYEE_ID, USER_NAME AS user_name,  TO_NUMBER(SUPPLIER_ID) as SUPPLIER_ID,
session_number as session_number,CUSTOMER_ID as CUSTOMER_ID, DESCRIPTION as
description, EMAIL_ADDRESS as EMAIL_ADDRESS, FAX as FAX, START_DATE  AS
START_DATE, END_DATE AS END_DATE,'Supplier' as PARTY_TYPE,b.person_first_name as
party_first_name,b.person_last_name as party_last_name,b.party_id as
party_id ,b.party_name as supplier_name,b.sup_party_id as supplier_party_id,null
as security_group_id, null as responsibility_ID,null as RESPONSIBILITY_APP_ID,
null AS RESP_END_DATE,null AS RESP_START_DATE,null as ROLE_ID ,null as
role_start_date ,null as expiration_date ,null as RESP_DESCRIPTION,null as
ROLE_APP_ID  from fnd_user a,(   select
hp.party_id,hp.person_first_name,hp.person_last_name,hp1.party_name,hp1.party_id
as sup_party_id FROM hz_relationships hr , hz_parties hp,hz_parties hp1 where
```

```
hr.subject_ID = hp1.party_id and hr.object_ID=hp.party_id and
hr.subject_type='ORGANIZATION' and hr.object_type='PERSON') b where
a.person_party_id= b.party_id \
               union all \
               select USER_ID AS user_id,USER_GUID AS user_guid,sysdate  as
system_date, LAST_UPDATE_DATE  DATE_UPDATED, case when password_lifespan_days >
0 then 'Days' when password_lifespan_accesses > 0 then 'Accesses' else 'None'
end as PASSWORD_EXP_TYPE, case when password_lifespan_days > 0 then
password_lifespan_days when password_lifespan_accesses > 0 then
password_lifespan_accesses else null end as PASSWORD_LIFESPAN, EMPLOYEE_ID as
EMPLOYEE_ID, USER_NAME AS user_name,  TO_NUMBER(SUPPLIER_ID) as SUPPLIER_ID,
session_number as session_number,CUSTOMER_ID as CUSTOMER_ID, DESCRIPTION as
description, EMAIL_ADDRESS as EMAIL_ADDRESS, FAX as FAX, START_DATE AS
START_DATE,END_DATE AS END_DATE,'Party' as PARTY_TYPE,b.person_first_name as
party_first_name,b.person_last_name as party_last_name,b.party_id as
party_id,null as supplier_name,null as supplier_party_id,null as
security_group_id, null as responsibility_ID,null as RESPONSIBILITY_APP_ID, null
AS RESP_END_DATE,null AS RESP_START_DATE ,null as ROLE_ID ,null as
role_start_date ,null as expiration_date ,null as RESP_DESCRIPTION,null as
ROLE_APP_ID from fnd_user a,(   select
hp.party_id,hp.person_first_name,hp.person_last_name  FROM hz_parties hp) b
where a.person_party_id not in (select hr.object_ID FROM hz_relationships hr
where hr.subject_type='ORGANIZATION' and hr.object_type='PERSON') and
a.person_party_id= b.party_id \
               union all \
               select USER_ID AS user_id,USER_GUID AS user_guid,sysdate as
system_date, LAST_UPDATE_DATE  DATE_UPDATED, case when password_lifespan_days >
0 then 'Days' when password_lifespan_accesses > 0 then 'Accesses' else 'None'
end as PASSWORD_EXP_TYPE, case when password_lifespan_days > 0 then
password_lifespan_days when password_lifespan_accesses > 0 then
password_lifespan_accesses else null end as PASSWORD_LIFESPAN, EMPLOYEE_ID as
EMPLOYEE_ID, USER_NAME AS user_name,  TO_NUMBER(SUPPLIER_ID) as SUPPLIER_ID,
session_number as session_number,CUSTOMER_ID as CUSTOMER_ID, DESCRIPTION as
description, EMAIL_ADDRESS as EMAIL_ADDRESS, FAX as FAX, START_DATE AS
START_DATE,END_DATE  AS END_DATE,null as PARTY_TYPE,null as
party_first_name,null as party_last_name,null as party_id,null as
supplier_name,null as supplier_party_id,null as security_group_id, null as
responsibility_ID,null as RESPONSIBILITY_APP_ID, null AS RESP_END_DATE,null AS
RESP_START_DATE ,null as ROLE_ID ,null as role_start_date ,null as
expiration_date ,null as RESP_DESCRIPTION,null as ROLE_APP_ID from fnd_user a
where person_party_id IS NULL \
               ) \
               select * from ( \
               select RESULTTABLE.*,ROW_NUMBER() OVER (ORDER BY user_id) AS
Row_Num from \
               ( \
               select * from party \
               union all \
               select f.USER_ID AS user_id,f.user_guid as
user_guid,f.system_date,f.DATE_UPDATED,f.PASSWORD_EXP_TYPE,f.PASSWORD_LIFESPAN,f.
EMPLOYEE_ID, f.USER_NAME,f.SUPPLIER_ID, f.session_number,f.CUSTOMER_ID,
f.DESCRIPTION , f.EMAIL_ADDRESS , f.FAX , f.START_DATE,f.END_DATE,f.PARTY_TYPE,
f.party_first_name,f.party_last_name,f.party_id,f.supplier_name,f.supplier_party_
id,s.security_group_id as security_group_id, (CONCAT(a.application_ID || '~',
r.responsibility_id)) as responsibility_id,ur.RESPONSIBILITY_APPLICATION_ID as
RESPONSIBILITY_APP_ID,ur.END_DATE  AS RESP_END_DATE,ur.START_DATE  AS
RESP_START_DATE, null as ROLE_ID ,null as role_start_date ,null as
expiration_date ,ur.DESCRIPTION as RESP_DESCRIPTION,null as ROLE_APP_ID from
party f,FND_USER_RESP_GROUPS_DIRECT ur, fnd_application_vl a,
fnd_responsibility_vl r, fnd_security_groups_vl s where f.user_id = ur.user_id
and ur.responsibility_ID = r.responsibility_ID and r.application_ID =
```

```
a.application_ID and ur.security_group_id = s.security_group_id and
( (ur.START_DATE < nvl(ur.END_DATE, TO_DATE('31-DEC-4712','dd-mon-yyyy')) and
ur.START_DATE > sysdate) or sysdate between ur.START_DATE and nvl(ur.END_DATE,
TO_DATE('31-DEC-4712','dd-mon-yyyy')) ) \
                union all \
                select f.USER_ID AS user_id,f.user_guid as
user_guid,f.system_date,f.DATE_UPDATED,f.PASSWORD_EXP_TYPE,f.PASSWORD_LIFESPAN,f.
EMPLOYEE_ID, f.USER_NAME,f.SUPPLIER_ID, f.session_number,f.CUSTOMER_ID,
f.DESCRIPTION , f.EMAIL_ADDRESS , f.FAX , f.START_DATE,f.END_DATE,f.PARTY_TYPE,
f.party_first_name,f.party_last_name,f.party_id,f.supplier_name,f.supplier_party_
id, null as security_group_id, null as responsibility_id, null as
RESPONSIBILITY_APP_ID, null AS RESP_END_DATE,null AS RESP_START_DATE,
(CONCAT(r.application_id || '~',   r.name)) AS ROLE_ID ,r.active_from  AS
role_start_date,r.active_to  AS expiration_date,null as
RESP_DESCRIPTION,r.application_id as ROLE_APP_ID  from party f , roledata r
where  f.user_name = r.user_name \
                ) RESULTTABLE  \
                --<FILTER> \
                )  WHERE Row_Num BETWEEN <START_ROW_NUMBER> and <END_ROW_NUMBER>
```

A.2 Sample SQL Queries Updated to Include Multivalued Attributes

Use this SQL query to update the UM_USER_RECON and UM_USER_SYNC queries.

The following SQL query can be used to update the UM_USER_RECON and UM_USER_SYNC queries if you have added new security attributes as part of performing the procedure described in Updating the Connector Bundle:

```
with roledata as ( \
                select
fa.application_id,fa.application_short_name,wlr.name,ura.user_name
user_name ,ura.start_date active_from,ura.end_date active_to from wf_local_roles
wlr,wf_user_role_assignments ura,fnd_application fa where   ura.role_name like
'UMX%' AND wlr.parent_orig_system = 'UMX' and  wlr.name=ura.role_name and
fa.application_short_name = wlr.owner_tag and  ( (ura.start_date <
nvl(ura.end_date, TO_DATE('31-DEC-4712','dd-mon-yyyy')) and ura.start_date >
sysdate) or sysdate between ura.start_date and nvl(ura.end_date, TO_DATE('31-
DEC-4712','dd-mon-yyyy')) ) \
                ) ,  securitydata as ( \
                select userak.web_user_id as user_id,userak.attribute_code as
SECURITY_ATTR_NAME, userak.attribute_application_id as
SECURITY_APP_ID,NVL(userak.VARCHAR2_VALUE,NVL(to_char(userak.DATE_VALUE),userak.N
UMBER_VALUE)) as SECURITY_ATTR_VALUE,ak.DATA_TYPE as SECURITY_ATTR_TYPE from
ak_web_user_sec_attr_values userak,AK_ATTRIBUTES ak where
ak.attribute_code=userak.attribute_code and ak.attribute_application_id=
userak.attribute_application_id \
                ), party as ( \
                select null as SECURITY_ATTR_NAME, null as SECURITY_APP_ID, null
as SECURITY_ATTR_VALUE,null as SECURITY_ATTR_TYPE,USER_ID AS user_id,USER_GUID
AS user_guid,sysdate  as system_date,    LAST_UPDATE_DATE  DATE_UPDATED, case
when password_lifespan_days > 0 then 'Days' when password_lifespan_accesses > 0
then 'Accesses' else 'None' end as PASSWORD_EXP_TYPE, case when
password_lifespan_days > 0 then password_lifespan_days when
password_lifespan_accesses > 0 then password_lifespan_accesses else null end as
PASSWORD_LIFESPAN, EMPLOYEE_ID as  EMPLOYEE_ID, USER_NAME AS user_name,
TO_NUMBER(SUPPLIER_ID) as   SUPPLIER_ID, session_number as
session_number,CUSTOMER_ID as CUSTOMER_ID, DESCRIPTION as description,
```

```
EMAIL_ADDRESS as EMAIL_ADDRESS, FAX as FAX, START_DATE  AS START_DATE, END_DATE
AS END_DATE,'Supplier' as PARTY_TYPE,b.person_first_name as
party_first_name,b.person_last_name as party_last_name,b.party_id as
party_id ,b.party_name as supplier_name,b.sup_party_id as supplier_party_id,null
as security_group_id, null as responsibility_ID,null as RESPONSIBILITY_APP_ID,
null AS RESP_END_DATE,null AS RESP_START_DATE,null as ROLE_ID ,null as
role_start_date ,null as expiration_date ,null as RESP_DESCRIPTION,null as
ROLE_APP_ID  from fnd_user a,(   select
hp.party_id,hp.person_first_name,hp.person_last_name,hp1.party_name,hp1.party_id
as sup_party_id FROM hz_relationships hr , hz_parties hp,hz_parties hp1 where
hr.subject_ID = hp1.party_id and hr.object_ID=hp.party_id
 and hr.subject_type='ORGANIZATION' and hr.object_type='PERSON') b where
a.person_party_id= b.party_id \
               union all \
               select null as SECURITY_ATTR_NAME, null as SECURITY_APP_ID, null
as SECURITY_ATTR_VALUE,null as SECURITY_ATTR_TYPE,USER_ID AS user_id,USER_GUID
AS user_guid,sysdate  as system_date,    LAST_UPDATE_DATE  DATE_UPDATED, case
when password_lifespan_days > 0 then 'Days' when password_lifespan_accesses > 0
then 'Accesses' else 'None' end as PASSWORD_EXP_TYPE, case when
password_lifespan_days > 0 then password_lifespan_days when
password_lifespan_accesses > 0 then password_lifespan_accesses else null end as
PASSWORD_LIFESPAN, EMPLOYEE_ID as  EMPLOYEE_ID, USER_NAME AS user_name,
TO_NUMBER(SUPPLIER_ID) as   SUPPLIER_ID, session_number as
session_number,CUSTOMER_ID as CUSTOMER_ID,    DESCRIPTION as description,
EMAIL_ADDRESS as EMAIL_ADDRESS, FAX as FAX, START_DATE  AS START_DATE,END_DATE
AS END_DATE,'Party' as PARTY_TYPE,b.person_first_name as
party_first_name,b.person_last_name as party_last_name,b.party_id as
party_id,null as supplier_name,null as supplier_party_id,null as
security_group_id, null as responsibility_ID,null as RESPONSIBILITY_APP_ID, null
AS RESP_END_DATE,null AS RESP_START_DATE ,null as ROLE_ID ,null as
role_start_date ,null as expiration_date ,null as RESP_DESCRIPTION,null as
ROLE_APP_ID from fnd_user a,(   select
hp.party_id,hp.person_first_name,hp.person_last_name  FROM hz_parties hp) b
where a.person_party_id not in (select hr.object_ID FROM hz_relationships hr
where hr.subject_type='ORGANIZATION' and hr.object_type='PERSON') and
a.person_party_id= b.party_id \
               union all \
               select null as SECURITY_ATTR_NAME, null as SECURITY_APP_ID, null
as SECURITY_ATTR_VALUE,null as SECURITY_ATTR_TYPE,USER_ID AS user_id,USER_GUID
AS user_guid,sysdate  as system_date,    LAST_UPDATE_DATE  DATE_UPDATED, case
when password_lifespan_days > 0 then 'Days' when password_lifespan_accesses > 0
then 'Accesses' else 'None' end as PASSWORD_EXP_TYPE, case when
password_lifespan_days > 0 then password_lifespan_days when
password_lifespan_accesses > 0 then password_lifespan_accesses else null end as
PASSWORD_LIFESPAN, EMPLOYEE_ID as  EMPLOYEE_ID, USER_NAME AS user_name,
TO_NUMBER(SUPPLIER_ID) as   SUPPLIER_ID, session_number as
session_number,CUSTOMER_ID as CUSTOMER_ID,    DESCRIPTION as description,
EMAIL_ADDRESS as EMAIL_ADDRESS, FAX as FAX, START_DATE  AS START_DATE,END_DATE
AS END_DATE,null as PARTY_TYPE,null as party_first_name,null as
party_last_name,null as party_id,null as supplier_name,null as
supplier_party_id,null as security_group_id, null as responsibility_ID,null as
RESPONSIBILITY_APP_ID, null AS RESP_END_DATE,null AS RESP_START_DATE ,null as
ROLE_ID ,null as role_start_date ,null as expiration_date ,null as
RESP_DESCRIPTION,null as ROLE_APP_ID from fnd_user a where person_party_id IS
NULL \
               ) \
               select * from ( \
               select RESULTTABLE.*,ROW_NUMBER() OVER (ORDER BY user_id) AS
Row_Num from \
               ( \
               select * from party \
```

```
                union all \
                select f.SECURITY_ATTR_NAME, f.SECURITY_APP_ID,
f.SECURITY_ATTR_VALUE,f.SECURITY_ATTR_TYPE,f.USER_ID AS user_id,f.user_guid as
user_guid,f.system_date,f.DATE_UPDATED,f.PASSWORD_EXP_TYPE,f.PASSWORD_LIFESPAN,f.
EMPLOYEE_ID, f.USER_NAME,f.SUPPLIER_ID, f.session_number,f.CUSTOMER_ID,
f.DESCRIPTION , f.EMAIL_ADDRESS , f.FAX , f.START_DATE,f.END_DATE,f.PARTY_TYPE,
f.party_first_name,f.party_last_name,f.party_id,f.supplier_name,f.supplier_party_
id,s.security_group_id as security_group_id, (CONCAT(a.application_ID || '~',
r.responsibility_id)) as responsibility_id,ur.RESPONSIBILITY_APPLICATION_ID as
RESPONSIBILITY_APP_ID,ur.END_DATE  AS RESP_END_DATE,ur.START_DATE  AS
RESP_START_DATE, null as ROLE_ID ,null as role_start_date ,null as
expiration_date ,ur.DESCRIPTION as RESP_DESCRIPTION,null as ROLE_APP_ID from
party f,FND_USER_RESP_GROUPS_DIRECT ur, fnd_application_vl a,
fnd_responsibility_vl r, fnd_security_groups_vl s where f.user_id = ur.user_id
and ur.responsibility_ID = r.responsibility_ID and r.application_ID =
a.application_ID and ur.security_group_id = s.security_group_id and
( (ur.START_DATE < nvl(ur.END_DATE, TO_DATE('31-DEC-4712','dd-mon-yyyy')) and
ur.START_DATE > sysdate) or sysdate between ur.START_DATE and nvl(ur.END_DATE,
TO_DATE('31-DEC-4712','dd-mon-yyyy')) ) \
                union all \
                select f.SECURITY_ATTR_NAME, f.SECURITY_APP_ID,
f.SECURITY_ATTR_VALUE,f.SECURITY_ATTR_TYPE,f.USER_ID AS user_id,f.user_guid as
user_guid,f.system_date,f.DATE_UPDATED,f.PASSWORD_EXP_TYPE,f.PASSWORD_LIFESPAN,f.
EMPLOYEE_ID, f.USER_NAME,f.SUPPLIER_ID, f.session_number,f.CUSTOMER_ID,
f.DESCRIPTION , f.EMAIL_ADDRESS , f.FAX , f.START_DATE,f.END_DATE,f.PARTY_TYPE,
f.party_first_name,f.party_last_name,f.party_id,f.supplier_name,f.supplier_party_
id, null as security_group_id, null as responsibility_id, null as
RESPONSIBILITY_APP_ID, null AS RESP_END_DATE,null AS RESP_START_DATE,
(CONCAT(r.application_id || '~',   r.name)) AS ROLE_ID ,r.active_from  AS
role_start_date,r.active_to  AS expiration_date,null as
RESP_DESCRIPTION,r.application_id as ROLE_APP_ID  from party f , roledata r
where  f.user_name = r.user_name \
                union all \
                select sa.SECURITY_ATTR_NAME as SECURITY_ATTR_NAME,
sa.SECURITY_APP_ID as SECURITY_APP_ID, sa.SECURITY_ATTR_VALUE as
SECURITY_ATTR_VALUE,sa.SECURITY_ATTR_TYPE as SECURITY_ATTR_TYPE,f.USER_ID AS
user_id,f.user_guid as
user_guid,f.system_date,f.DATE_UPDATED,f.PASSWORD_EXP_TYPE,f.PASSWORD_LIFESPAN,f.
EMPLOYEE_ID, f.USER_NAME,f.SUPPLIER_ID, f.session_number,f.CUSTOMER_ID,
f.DESCRIPTION , f.EMAIL_ADDRESS , f.FAX , f.START_DATE,f.END_DATE,f.PARTY_TYPE,
f.party_first_name,f.party_last_name,f.party_id,f.supplier_name,f.supplier_party_
id, null as security_group_id, null as responsibility_id, null as
RESPONSIBILITY_APP_ID, null AS RESP_END_DATE,null AS RESP_START_DATE, null AS
ROLE_ID ,null  AS role_start_date,null  AS expiration_date,null as
RESP_DESCRIPTION,null as ROLE_APP_ID  from party f , securitydata sa where
f.user_id = sa.user_id \
                ) RESULTTABLE  \
                --<FILTER> \
                )  WHERE Row_Num BETWEEN <START_ROW_NUMBER> and <END_ROW_NUMBER>
```

B

Sample Code Snippets for Extending the Connector Schema

This appendix lists sample code snippets for extending the connector schema by adding a multivalued attribute (for example, __SECUTIRY_ATTRS__). All the code snippets listed in this appendix consider __SECURITY_ATTRS__ as the multivalued attribute being added to the connector schema.

The following is a sample code snippet for extending the connector schema to include the multivalued attribute that has been initialized by specifying the number of child attributes:

```
        attr := attributelist();
        attr.extend(5);
        attr (1) := attributeinfo('SECURITY_ATTR_NAME','varchar',1,1,1,1);
        attr (2) := attributeinfo('SECURITY_ATTR_VALUE','varchar',1,1,1,1);
        attr (3) := attributeinfo('SECURITY_ATTR_TYPE','varchar',1,1,1,1);
        attr (4) := attributeinfo('SECURITY_APP_ID','varchar',1,1,1,1);

        schemaout.extend;
    schemaout( 4 ) := schema_object('__SECURITY_ATTRS__',attr);
```

The following is a sample code snippet for extending the connector schema to include the multivalued attribute without initializing the child attributes in advance:

```
        attr := attributelist();
        attr.extend;
        attr (1) := attributeinfo('SECURITY_ATTR_NAME','varchar',1,1,1,1);
        attr.extend;
        attr (2) := attributeinfo('SECURITY_ATTR_VALUE','varchar',1,1,1,1);
        attr.extend;
        attr (3) := attributeinfo('SECURITY_ATTR_TYPE','varchar',1,1,1,1);
        attr.extend;
        attr (4) := attributeinfo('SECURITY_APP_ID','varchar',1,1,1,1);

        schemaout.extend;
    schemaout( 4 ) := schema_object('__SECURITY_ATTRS__',attr);
```

The following is a sample code snippet for extending the connector schema to include the multivalued attribute with mixed ways of initializing the child attributes:

```
        attr := attributelist();
        attr.extend(2);
        attr (1) := attributeinfo('SECURITY_ATTR_NAME','varchar',1,1,1,1);
        attr (2) := attributeinfo('SECURITY_ATTR_VALUE','varchar',1,1,1,1);
        attr.extend;
        attr (3) := attributeinfo('SECURITY_ATTR_TYPE','varchar',1,1,1,1);
        attr.extend;
        attr (4) := attributeinfo('SECURITY_APP_ID','varchar',1,1,1,1);

        schemaout.extend;
    schemaout( 4 ) := schema_object('__SECURITY_ATTRS__',attr);
```

C

Files and Directories in the EBS User Management Connector Package

These are the files and directories on the connector installation package that comprise the EBS User Management connector.

Table C-1 Files and Directories in the Installation Package

File in the Installation Package Directory	Description
bundle/org.identityconnectors.ebs-12.3.0.jar	This JAR file contains the connector bundle.
configuration/EBS-UM-CI.xml	This file is used for installing a CI-based connector. This XML file contains configuration information that is used by the Connector Installer during connector installation.
resources/EBS-UM.properties	This file is a resource bundle that contains language-specific information that is used by the connector. During connector installation, these resource bundles are copied to the Oracle Identity Manager database. **Note:** A **resource bundle** is a file containing localized versions of the text strings that are displayed on the Administrative and User Console. These text strings include GUI element labels and messages.
scripts/OIM_FND_GLOBAL.pck	This package contains the procedures that are called to initialize the global security context for a database session during provisioning operations.
scripts/OIM_FND_USER_TCA_PKG.pck	This is a customized wrapper package for creating and updating party records.
scripts/GET_LAST_UPDATE_DATE_FUNCTION.pck	This package contains the procedures that gets latest date from roles and responsibilities.
scripts/OIM_EBSUM_SCHEMA_PKG.pck	This package contains the procedures that generates schema for connector.
scripts/OIM_TYPES.pck	This package file contains SQL statements used for creating Oracle types. Oracle types are used for storing OIG schema.
scripts/OimUser.sql scripts/OimUserAppstablesSynonyms.sql scripts/OimUserGrants.sql scripts/OimUserSynonyms.sql scripts/OimUserAD_ZDGrants.sql	These files contain the SQL scripts to create a target system user account, grant the required rights to the user, and create synonyms of various database objects to be used by the connector. See Creating a Target System User Account for Connector Operations for more information about this user.
scripts/Run_UM_DBScripts.bat scripts/Run_UM_DBScripts.sh	This file contains commands to run the SQL scripts for creating a service account with the required grants. See Creating a Target System User Account for Connector Operations for more information about this user.
upgrade/PostUpgradeScript_PlainEBSUM.sql	This file is used during the plain Oracle EBS User Management connector upgrade procedure.

Table C-1 (Cont.) Files and Directories in the Installation Package

File in the Installation Package Directory	Description
upgrade/PostUpgradeScript_TCAEBSUM.sql	This file is used during Oracle EBS User Management TCA connector upgrade procedure.
xml/EBS-UM-ConnectorConfig.xml xml/EBSUM-pre-config.xml xml/EBSUM-target-template.xml	This XML file contains definitions for the following connector components: • Resource objects • IT resource types • IT resource instance • Process forms • Process tasks and adapters • Process definition • Prepopulate rules • Lookup definitions • Reconciliation rules • Scheduled tasks