Oracle® Fusion Middleware

System Administrator's Guide for Oracle Identity Manager

11*g* Release 1 (11.1.1)

E14308-05

February 2011

ORACLE®

Oracle Fusion Middleware System Administrator's Guide for Oracle Identity Manager, 11*g* Release 1 (11.1.1)

E14308-05

Primary Author: Debapriya Datta

Contributing Authors: Prakash Hulikere, Alankrita Prakash,, Devanshi Mohan, Lyju Vadassery

Contributor: Sid Choudhury

Contents

20 Installing and Configuring a Remote Manager

21 Using the Form Version Control Utility

22 Using the Archival Utilities

Part V Performance Tuning and Best Practices

23 Tuning Oracle Database

24 Tuning Application Server Performance

25 Tuning Connector Performance

List of Examples

List of Figures

List of Tables

Preface

The *Oracle Fusion Middleware System Administrator's Guide for Oracle Identity Manager* describes how to perform system administration tasks in Oracle Identity Manager.

Audience

This guide is intended for system administrators who can perform system configuration tasks such as horizontal migration of system configuration, performance tuning across database, application servers, JMS, and connectors, scheduled task management, connector installation and deployment, and archival utility management.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Related Documents

For more information, refer to the following documents:

- *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*
- *Oracle Fusion Middleware User's Guide for Oracle Identity Manager*
- *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*
- *Oracle Fusion Middleware Quick Installation Guide for Oracle Identity Management*
- *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management*
- *Oracle Fusion Middleware Suite Integration Overview*
- *Oracle Fusion Middleware User Reference for Oracle Identity Management*
- *Oracle Fusion Middleware High Availability Guide*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Access Manager*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Adaptive Access Manager*
- *Oracle Fusion Middleware Developer's Guide for Oracle Adaptive Access Manager*
- *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Navigator*
- *Oracle Fusion Middleware Administrator's Guide for Authorization Policy Manager*

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.

Part I

Application Management

This part describes the application management tasks in Oracle Identity Manager.

It contains the following chapters:

1

Managing Reconciliation Events

The reconciliation process involves generation of events to be applied to Oracle Identity Manager. These events reflect atomic changes in the target system, and contain the data that has changed, the type of change, along with other information. The reconciliation events that are generated as a result of changes occurring in the target system must be managed in such a way that they meet various business requirements. The Event Management section in the Oracle Identity Manager Advanced Administration addresses these event management requirements.

> **See Also:** "Reconciliation Configuration" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for detailed information about reconciliation

You can manage reconciliation events by using the Event Management section, which lets you query the events stored in various ways and display all event data. The events are always displayed in the same form, which is on the Event Details page. You can run custom queries for the events through the Advanced Search feature. It also allows you to perform any necessary action to resolve event issues.

Events are generated by reconciliation runs, which are scheduled to run by using the Oracle Identity Manager Scheduler.

> **See Also:** ""Managing Scheduled Tasks" on page 2-1" for detailed information about the scheduler

This chapter describes the following topics:

- Reconciliation Features in Oracle Identity Manager
- Event Management Tasks
- Updating Reconciliation Profiles Manually
- Populating Data in the RECON_EXCEPTIONS Table

1.1 Reconciliation Features in Oracle Identity Manager

Reconciliation features can be divided into the following categories:

- Performance Enhancements
- Web-Based Event Management Interface
- Other Enhancements

1.1.1 Performance Enhancements

In 11*g* Release 1 (11.1.1), the following enhancements help increase performance during reconciliation:

- New Metadata Model - Profiles
- Parameters to Control Flow and Processing of Events
- Grouping of Events by Reconciliation Runs
- Grouping of Events by Batches
- Implementing Reconciliation Engine Logic in the Database
- Improved Java Engine
- Improved Database Schema

1.1.1.1 New Metadata Model - Profiles

Oracle Identity Manager has a new model to store the metadata associated with various targets.

In earlier releases, the metadata is associated with a reconciliation target. This limits the ability to run multiple jobs performing different types of reconciliation against the same target.

In Oracle Identity Manager 11*g* Release 1 (11.1.1), all configurations in various components of Oracle Identity Manager are stored centrally in an XML store called MDS.

For backward compatibility, current deployments continue managing their configurations through Oracle Identity Manager Design Console and the configuration continues to be stored in the Oracle Identity Manager database. The configuration APIs automatically read the configurations from the tables in Oracle Identity Manager 11*g* Release 1 (11.1.1) and convert them into XML profiles, called default profiles, and associate those profiles with the existing reconciliation runs. The default profiles are marked with a DEFAULT tag.

You manage all the metadata by using Oracle Identity Manager Design Console. Using Oracle Identity Manager Design Console, you can generate the default reconciliation profile. This can be used to regenerate the profile when reconciliation configurations are changed from Oracle Identity Manager Design Console. When configurations are imported from the Deployment Manager, the profile is generated by default.

All nondefault profiles are completely managed directly by using an XML editor.

> **See Also:** "Reconciliation Profile" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about reconciliation profiles

1.1.1.2 Parameters to Control Flow and Processing of Events

This section consists of the following topics:

- Parameters to Control Event Processing
- Parameter to Control AutoRetry

Parameters to Control Event Processing

BatchSize is the parameter to control event processing. This dictates the size of the batch. A batch size of 1 is equivalent to processing of events one at a time. Batch size is

available as a system property and can be managed from Oracle Identity Manager Design Console. The property name is OIM.ReconBatchSize. The default value of the system BatchSize parameter is 500. For information about system properties, see Chapter 4, "Administering System Properties".

Parameter to Control AutoRetry

The MaxRetryCount profile parameter controls auto retry by indicating how many times an item needs to be retried before the reconciliation engine marks it as an error or sends it to manual queue. MaxRetryCount = 0 means auto retry option is not configured.

> **See Also:** "Handling of Race Conditions" on page 1-5 for more information about auto retry

1.1.1.3 Grouping of Events by Reconciliation Runs

All the events created in the reconciliation database are grouped by reconciliation runs. All events in a reconciliation run are grouped with a common reconciliation run ID. Because each reconciliation run is associated with a profile, all events in a reconciliation run are processed by using the same profile. This helps in optimizing the performance because the configurations have to be retrieved only once per reconciliation run.

Each profile can use a different batch size. This enhances system performance for each target reconciliation by tuning the appropriate batch for it.

1.1.1.4 Grouping of Events by Batches

Batches are introduced to increase system performance during reconciliation. A batch consists of a number of events. It is a unit of processing in the reconciliation engine. The size of the batch is configurable. Reconciliation runs are broken into fixed size batches. For example, if a reconciliation run consists of 9900 events and batch size is 1000, then that reconciliation run is divided into 10 batches each with size 1000, and last batch with size 900.

Processing a batch as a unit optimizes system performance by eliminating the overhead of processing one event at a time. This also allows performing bulk operations wherever possible. Batches can also run in parallel to balance the use of hardware resources.

1.1.1.5 Implementing Reconciliation Engine Logic in the Database

In earlier releases, all engine logic was implemented in Java and the processing happened one event at a time. In 11*g* Release 1 (11.1.1), most of the logic to process the events is implemented as stored procedures. A combination for processing at batch level and the logic being implemented in PLSQL makes it possible to perform bulk operations at the SQL layer. The following steps are performed in bulk (one batch at a time):

- Required data check
- Applying matching rules
- Applying action rules

1.1.1.6 Improved Java Engine

Processing that cannot be performed in stored procedures and must be performed in Java layer also provides better performance than earlier releases of the engine for the following reasons:

- Java engine performs bulk operations by default:

  Submits events in batches to the database

  Submits bulk postprocess orchestration depending on the action

- Performs bulk operations wherever possible.

1.1.1.7 Improved Database Schema

A notable performance enhancement from the new database schema in 11*g* Release 1 (11.1.1) is by using horizontal tables for storing event details for various targets instead of using a single vertical table for storing the event details from various targets. A horizontal table is used for each profile.

> **See Also:** "Horizontal Tables" on page 1-4 for more information about horizontal tables

1.1.2 Web-Based Event Management Interface

Oracle Identity Manager provides a Web-based event management interface that allows you to manage the events from the Web. Authorized users are able to search for events, users, and handle exceptions by linking events with users and accounts. You can also close events, force failed events to be re-evaluated, and perform ad-hoc linking.

Ad-hoc linking refers to the ability provided to authorized users of the Event Management section to link an event to any user in Oracle Identity Manager. Although the reconciliation engine finds user matches for events, the user through this ad-hoc link feature can ignore those matches and select a different user. This allows you to handle exceptions resulting from error matches.

> **See Also:** "Event Management Tasks" on page 1-7 for information about the tasks performed in Event Management

1.1.3 Other Enhancements

Other reconciliation enhancements are described in the following sections:

- Horizontal Tables
- Handling of Race Conditions
- OES Integration
- Ad Hoc Linking

1.1.3.1 Horizontal Tables

In earlier releases of Oracle Identity Manager, the reconciliation schema has one table to store all the event details from various targets. The list of attributes and their names and types that the various reconciliation events contain can vary from target to target. This means that events from one target can contain a different set of data compared to events from another target. The only way to store data from such events in a single table is by storing one attribute per row. Therefore, in earlier releases, each row in the event detail table represents a single attribute of reconciliation event data. For each

attribute, it stores the event to which it belongs, the attribute name, type, and value. This is also referred to as vertical table in this document. Although vertical tables are beneficial from the point of view of flexibility and extensibility, it is not an efficient way to store event records from the performance prospective.

In 11*g* Release 1 (11.1.1), storage in vertical tables is replaced by separate tables for each target, called horizontal tables. They are called horizontal tables because instead of storing attributes of an event vertically in the table as rows (as many rows as there are number of attributes), the attributes of an event are stored horizontally as columns. This means that there are as many columns as there are number of attributes for a target. Each event is stored as a row. Because different targets can have different sets of attributes, each target has a separate table in the reconciliation schema to store event details. There can be multiple tables per target because of requirements to handle multi-valued attributes that are stored as rows in child tables.

Each row of the event detail table for a specific profile stores the list of reconciliation fields for a single event. For example, for trusted user reconciliation in which firstname, lastname, email attributes are being reconciled, there is the RA_XELLERATE_USER horizontal table with the following columns:

RE_KEY, RECON_FIRSTNAME, RECON_LASTNAME, RECON_EMAI

Creating and Maintaining Horizontal Tables

Horizontal tables can be created only when a target is being deployed against Oracle Identity Manager. This is because, at the time of target deployment, the reconciliation system knows the list of attributes and their types for the target, which needs to be reconciled.

Horizontal tables are updated when configurations are imported from the Deployment Manager or changes are made by using Oracle Identity Manager Design Console. To generate a horizontal table from Oracle Identity Manager Design Console, in the Object Reconciliation form, click **Generate Reconciliation Profile**.

1.1.3.2 Handling of Race Conditions

In earlier releases of Oracle Identity Manager, when an event is being reconciled, the reconciliation engine may not be able to process it successfully because before this event can be reconciled, another event needs to be reconciled. For example, before the reconciliation engine can reconcile an event that is supposed to create an account, the engine needs to reconcile an event that is supposed to create a user. This is called a race condition.

In Oracle Identity Manager 11*g* Release 1 (11.1.1), the race conditions are handled by using an auto retry option that you can select for each reconciliation run. To configure auto retry, specify a value greater than 0 for the MaxRetryCount parameter. If you do not want to configure auto retry, then specify 0 as the value of the MaxRetryCount parameter.

Note: MaxRetryCount is a parameter in the reconciliation profile. The default value of this parameter is 5. You can change this by exporting the profile from MDS, updating the retry count, and importing it back to MDS. For information about manually updating reconciliation profiles, see "Updating Reconciliation Profiles" on page 1-15.

When auto retry is configured, the reconciliation engine checks for the race conditions. If a race condition is found, then the reconciliation engine puts the reconciliation event in a re-evaluate queue until the retry count is exhausted.

A Reconciliation Retry Scheduled Task periodically checks if there is any event waiting for retry and is ready to be re-evaluated and if yes, it queues them up for reconciliation engine processing. This scheduled task is configured by default.

Note:

- If the auto retry count is exhausted, the reconciliation engine does not further process the event and sets the status per the matching rules. However, you can manually retry by requesting for re-evaluate from Event Management. For information about re-evaluating events, see "Re-evaluating Events" on page 1-11.
- During the retry, if the event is successfully processed, then the value of the CurrentRetryCount parameter is reset to 0.

Auto retry can handle the following race conditions:

- An account event for creating an account in Oracle Identity Manager is processed before the user is created for this event because the event for creating user is not processed yet.
- A user event for creating a Xellerate user in Oracle Identity Manager is processed before the organization is created to which this user belongs.

> **See Also:** "Parameter to Control AutoRetry" on page 1-3 for information about auto retry parameters

Except for the CurrentRetryCount parameter, all other auto retry parameters are stored as part of the reconciliation profiles. This means that while the events belonging to one reconciliation run may have auto retry configured, the events belonging to another reconciliation run may not have auto retry configured.

In Oracle Identity Manager 11*g* Release 1 (11.1.1), there is no UI to manage these parameters within a profile and you must use an XML editor to manage them by directly editing the XML profile. For information about editing an XML profile, see "Updating Reconciliation Profiles" on page 1-15.

1.1.3.3 OES Integration

The event management APIs, the reconciliation APIs, and the UI to manage reconciliation events are protected by using authorization policies. Oracle Entitlements Server (OES) is the Oracle product that is used to control authorization policies.

Note: More information about OES is available in the following URL:

`http://www.oracle.com/technology/products/id_mgmt/oes/index.html`

The default authorization policy for reconciliation specifies that only users with the Reconciliation Administrator or System Administrator role are able to access and use reconciliation.

See Also:

- "Managing Authorization Policies" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for detailed information about authorization policies
- "Managing Roles" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about how to assign roles to a user

1.1.3.4 Ad Hoc Linking

If the reconciliation engine is not able to determine the owner based on the matching rules, then you can manually link an account to a user by using Oracle Identity Manager Advanced Administration. Subsequent modifications to the account is automatically linked to that account.

Ad hoc linking is supported for user and account events. If the reconciliation engine is not able to determine the owner based on the matching rules, then you can manually link a user or account event to a user.

See Also: "Ad Hoc Linking" on page 1-13 for information about how to perform ad hoc linking

1.2 Event Management Tasks

You can perform the following event management tasks by using the Event Management section of Oracle Identity Manager Advanced Administration:

- Searching Events
- Displaying Event Details
- Determining Event Actions
- Re-evaluating Events
- Closing Events
- Linking Reconciliation Events

1.2.1 Searching Events

You can display a summary of reconciliation events by performing the following types of search:

- Performing a Simple Search for Events
- Performing an Advanced Search for Events

1.2.1.1 Performing a Simple Search for Events

To perform a simple search for events:

1. Login to Oracle Identity Manager Advanced Administration.
2. In the Welcome page, under Event Management, click **Search Reconciliation Events**. Alternatively, you can click the **Event Management** tab, and then click **Reconciliation**.
3. In the left pane, enter a search criterion in the Search field. You can include wildcard characters (*) in your search criterion.

The simple search takes one argument. The text arguments are searched in the following event fields:

- Event ID
- Profile Name
- Key Fields

Note: In simple search, you cannot perform the search by event dates.

4. Click the icon next to the Search field. The events that match your search criterion is displayed in the search results table.

   The search fetches all rows for which the aforementioned attributes contains the string specified in the Search field. The search result displays the Event ID, Profile Name, and Key Fields columns. The Event ID column displays the event ID. The IDs are sorted as integers, not strings. The Profile Name column displays the name of the reconciliation profile. Key field is an attribute that uniquely identifies a row of data. In reconciliation, some attributes are flagged as Key in the profile. These fields are displayed in the Key Fields column.

   Note: Simple Search is paginated, meaning it only displays search results 64 rows at a time. This is to improve performance. Scrolling down past the 64th row in the UI triggers another page fetched from the database and so on for every 64 rows beyond that.

1.2.1.2 Performing an Advanced Search for Events

The advanced search takes multiple arguments and lets you fine-tune the list of events. To perform an advanced search for events:

1. In the left pane of the Reconciliation section, click **Advanced Search**. The Search: Events page is displayed.
2. Select any one of the following options:
   - **All:** On selecting this option, the search is performed with the AND condition. This means that the search operation is successful only when all the search criteria specified are matched.
   - **Any:** On selecting this option, the search is performed with the OR condition. This means that the search operation is successful when any search criterion specified is matched.
3. In the Event ID field, enter the event ID that you want to search. You can use wildcard characters (*) in your search criteria. Select a search condition in the list adjacent to the Event ID field.
4. Specify search arguments in the other fields displayed in the Search: Events page. Table 1 1 lists the fields in the Search: events page.

Table 1 1 Advanced Search Fields

Field	Description
Event Id	The event ID. The IDs are sorted as integers, not strings.

Table 1 1 (Cont.) Advanced Search Fields

Field	Description
Resource Name	The name of the resource object representing the target system the event originates from.
Current Status	A string representing the current state of the event.
Type	The type of operation performed by the event: regular (add or modify) or delete.
Profile Name	The name of the reconciliation profile this event pertains to. **See Also:** "Reconciliation Profile" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about reconciliation profile
Entity	The type of Oracle Identity Manager entity this event pertains to. Can be either user, account, role, role grant, or role hierarchy.
Start Date	Oldest event creation date to search for.
End Date	Most recent event creation date to search for.
Linked User Login	A string representing the login ID of the user linked to the event.
Key Fields	The fields flagged as key fields in the reconciliation profile that uniquely identifies rows of data.

5. Click **Search**. The search results are displayed, which consists of the Event ID, Resource Name, Entity, Event Status, Type, Profile Name, Job ID, Key Fields, and Date columns.

From the search results, you can perform event bulk actions, such as close and re-evaluate, and also display the details of any specific event.

If you want to search for events with LDAP profile, use the following LDAP profiles in your search:

Object	Profile
User	LDAPUser
Role	LDAPRole
Role Membership	LDAPRoleMembership
Role Hierarchy	LDAPRoleHierarchy

1.2.2 Displaying Event Details

To display the details pertaining to an event:

- In the left pane of the Oracle Identity Manager Advanced Administration, from the list of events, select an event whose details you want to display.
- From the advanced search result table, click an event in the Event ID column.
- From the Actions list, select **Lookup**. The Event Details page is displayed. The fields in the Event Details page change dynamically based on the event type and event status. Alternatively, you can select an event from the Event Summary on the right pane, and click the magnifying glass icon for lookup to open the Event Details page.

The data in the Event Details page is displayed in the following sections:

- **Event:** This section displays the information about the event, such as event ID, whether the event type is User or Account, the time when the event was created, the reconciliation run ID, resource name, the profile name, and the key field values. Reconciliation can use several key fields, and the key field values are shown separated by commas.
- **Linked To:** This section shows that the event is linked to a user or account. It displays the user or account ID to which the event is linked, the account description (if any), and the type of linking, such as rule-based linking or manual linking. Rule-based linking means that the reconciliation engine has performed the linking. Manual linking means that the administrator performs the linking manually.
- **Notes:** The reconciliation engine adds notes where appropriate. For example, when there is a 'Data Validation Fail', the engine adds a note explaining the reason. This is a read-only field and is blank if no notes are attached to the event.
- **Reconciliation Data:** This table displays the reconciliation event data. This shows the attribute name, attribute value, and Oracle Identity Manager mapped field. It also shows the child data of the event, if any. The reconciliation data displays the last name, first name, hiring date, user ID, and the IT resource name.

  If there are attributes with multi-language support, then these attribute values are also displayed in a separate table similar to child data.
- **Matched Accounts:** This table displays the accounts that are matched. The columns in the Matched Accounts table are listed in Table 1 2:

Table 1 2 Columns in the Matched Accounts Table

Column	Description
Account ID	The account ID of the matched account
Orc Key	An internal key that is stored in the ORC table. This key indicated if the event is matched to a user or an account.
Descriptor Field	A description that is associated to the account
Login ID	The user login ID corresponding to the user ID displayed for user events.
Account Owner Name	A string comprising of the first name and last name and the login ID of the user who owns the account. The event pertains to this account.
Account Owner Type	The type of account owner, such as user.

- **Matched Users:** This table shows the user matches found by the reconciliation engine. For a multiple match, the linked user is not shown in this table.
- **History:** This table shows the operations that took place for this event from event creation and data validation to account matching and whether the update was successful. The columns in the History table are listed in Table 1 3:

Table 1 3 Columns in the History Table

Column	Description
Status	Event status at the given date and time.
Action	Action performed on the event at the given date and time.
Action Performed by User	The ID and login ID of the user who performed the cited action. The engine uses the Default IAM Admin id: xelsysadm, ID = 1.

Table 1 3 (Cont.) Columns in the History Table

Column	Description
Date and Time	Date and time of the cited action.
Notes	Any notes attached to the event at the specified date and time.

Note: Oracle Identity Manager does not support translation of the reconciliation field names.

1.2.3 Determining Event Actions

The list of actions allowed for an event depends on the status, type, and operation of the event. Table 1 4 lists the possible actions for each type and status of events.

Table 1 4 Actions for Event Status and Types

Event Status	Event Type	Possible Actions
No matches found	User	Close event
		Re-apply reconciliation rules
		Create entity
		Ad-hoc linking
	Account	Close event
		Re-evaluate event
		Ad-hoc linking
Users matched	User	Close event
		Re-apply reconciliation rules
		Linking
	Account	Close event
		Re-apply reconciliation rules
		Linking
Accounts matched	Account	Close event
		Re-apply reconciliation rules
		Linking
Event Received	Any	Close event

The possible actions are described in the subsequent sections.

1.2.4 Re-evaluating Events

Re-evaluating an event means reapplying the reconciliation rules on the event. Reconciliation rule refers to the matching rule used to identify the owner of an event. For instance, if you change the reconciliation rules by using Oracle Identity Manager Design Console, then you can re-evaluate the rules in the Event Management section of the Oracle Identity Manager Advanced Administration.

To re-evaluate an event:

1. From the list of events, select an event. You can select multiple event rows by pressing the Ctrl key if you want to re-evaluate multiple events at a time.

2. From the Actions list, select **Re-Evaluate Event**. The Re-Evaluate Event dialog box is displayed with the event IDs that you have selected.
3. Click **Perform**. A confirmation message is displayed stating that the reconciliation rules are successfully reapplied for the event. If the selected action fails for any event, a generic message is displayed that shows the event IDs for which bulk processing has failed. The events can then be processed one at a time.

Note:

- The preprocess validation lists the events that are valid and those that are invalid for re-evaluation. If you click Perform, then only the valid events are re-evaluated.
- All event actions are tracked in the Event History table.

1.2.5 Closing Events

This action closes or discards the selected events, and the events are removed from any further processing queues. To close an event:

1. From the list of events, select an event.
2. From the Actions list, select **Close Event**. You can select multiple event rows by pressing the Ctrl key if you want to close multiple events at a time. The Close Event dialog box is displayed.

Note: If closing an event is not a valid option, then en error message is displayed in the Close Event dialog box.

3. In the Justification box, enter a reason to close the event.
4. Click **Perform**. A confirmation message is displayed stating that the event is closed. If the selected action fails for any event, a generic message is displayed that shows the event IDs for which bulk processing has failed. The events can then be processed one at a time.

Note:

- All event actions are tracked in the Event History table.
- The close event operation needs a justification to be entered. Therefore, when multiple events are closed at a time by performing bulk action, all the closed events will have the same justification.

1.2.6 Linking Reconciliation Events

Oracle Identity Manager allows you to perform the following operations for linking reconciliation events:

- Ad Hoc Linking
- Manual Linking
- Linking Orphan Accounts

1.2.6.1 Ad Hoc Linking

Ad hoc linking allows you to link an event to any user or role in Oracle Identity Manager. Even if the reconciliation engine finds user matches for the events, you can use ad hoc linking to ignore those matches and select a different user. This allows you to handle exceptions resulting from error matches because the reconciliation matching rules may not work correctly all the time.

This action lets you link an event to any entity other than the already matched entities. In other words, instead of selecting a row from the Matched Users table, you can select another user to link with the event.

To create an ad hoc link for an event:

1. In the Event Details page, from the Actions list, select **Ad Hoc Link**. The Ad Hoc Link dialog box is displayed.
2. Perform a user search by specifying a search criterion.
3. Select a user from the search result, and click **Perform**. A confirmation message is displayed that states that the ad hoc linking with the event is successful.

1.2.6.2 Manual Linking

When a reconciliation event has multiple matches, each match is displayed on the Matched Accounts (for account entity) or Matched Users (for user entity) tab of the Event Details page. You can manually select any match out of all the matches found by the reconciliation engine. To perform manual linking:

> **Note:** In manual linking, you select a match from a list of matches found by the reconciliation engine instead of selecting from a list of all Oracle Identity Manager users.

1. In the Event Details page, select a row from the table that lists all the matches found by the reconciliation engine.
2. Click **Link**. A message is displayed asking for confirmation.
3. Click OK to confirm.

1.2.6.3 Linking Orphan Accounts

Orphan accounts refer to accounts in the target system for which there is no corresponding user that exists in Oracle Identity Manager.

You can resolve events for orphan accounts for which the events either have no user match in Oracle Identity Manager, or several users are found for the match. You can therefore perform any one of the following:

- Re-create the user in Oracle Identity Manager
- Trigger a provisioning process to delete the user or account from the target system
- Perform ad hoc or manual linking

The Event Management section allows you to resolve orphan accounts by selecting the correct user for the match in the following scenarios:

- For an Event With Multiple Matches
- For an Event With No Matches

1.2.6.3.1 For an Event With Multiple Matches When several users are matched to the event data by the reconciliation engine, you must select the right user by using ad hoc or manual linking.

For information about ad hoc linking, see "Ad Hoc Linking" on page 1-13.

For information about manual linking, see "Manual Linking" on page 1-13.

1.2.6.3.2 For an Event With No Matches When no matches are found for an event, you can either trigger an entity creation, or select an Oracle Identity Manager entity to link to the event. For information about how to select and Oracle Identity Manager entity to link to an event, see "Ad Hoc Linking" on page 1-13.

1.3 Updating Reconciliation Profiles Manually

This section describes creating and updating reconciliation profiles manually in the following sections:

- Creating New Reconciliation Profiles
- Updating Reconciliation Profiles
- Changing the Profile Mode

1.3.1 Creating New Reconciliation Profiles

You might want to create reconciliation profiles in the following scenarios:

- Creating Additional Nondefault Profiles for Reconciliation Based on Resource Objects
- Creating New Profiles for Trusted Source Reconciliation

1.3.1.1 Creating Additional Nondefault Profiles for Reconciliation Based on Resource Objects

For reconciliation based on resource objects, the default profile name is the same as that of the resource object. For example, if resource object name is testresource, then the default profile name is also testresource. The corresponding reconciliation horizontal table name is RA_TESTRESOURCE<obj_key>. If the resource has Multi-Language Support (MLS) data, then the MLS table name is RA_MLS_TESTRESOURCE<obj_key>.

If the resource object has child tables, then for each child form name, which is UD_xxx, there is a corresponding RA_UD_xx. Each of the tables has a corresponding entity definition XML file, which is stored as per platform documentation on MDS storage. Therefore, RA_MLS_TESTRESOURCE<obj_key> has an entity definition MDS document called /db/RA_TESTRESOURCE<obj_key>.xml, which is stored as per platform documentation on MDS storage.

Note: If you change the name of a resource object, the reconciliation profile needs to be regenerated by clicking the "Create Reconciliation Profile" button in the Object Reconciliation tab in Oracle Identity Manager Design Console.

If the resource has child tables, then you must first delete all horizontal tables and entity definitions for the RA_UD_xx tables associated with the reconciliation profile, before regenerating it.

To create nondefault profiles for reconciliation based on resource objects:

> **Note:** You can export or import files to MDS by using the MDS export/import utility, which is run by running the weblogicExportMetadata.sh and weblogicImportMetadata.sh scripts. For information about running these scripts, see "MDS Utilities" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

1. Create a copy of the exported profile XML file with a different name.
2. Make changes to the file to reflect the new profile name, horizontal table names, and new reconciliation field names and mappings.
3. Import the new profile to MDS by using the MDS import tool.
4. Copy the entity definition XML files with new names based on the new profile name. If the reconciliation field names also change, then change the XML files to refer to the new reconciliation field names.
5. Import the entity definition XML files to MDS by using the MDS import tool.
6. Create new horizontal tables in the database based on the new profile name.

1.3.1.2 Creating New Profiles for Trusted Source Reconciliation

The procedure for creating new profiles for trusted source reconciliation is similar to the procedure in "Creating Additional Nondefault Profiles for Reconciliation Based on Resource Objects" on page 1-14. The only difference is that trusted source reconciliation may or may not be associated with a resource object, and therefore, you can use the XML files corresponding to the LDAPUser profile as samples.

1.3.2 Updating Reconciliation Profiles

To change a property in a reconciliation profile, for instance batch size:

1. Export the /db/PROFILE_NAME profile document from MDS.
2. Make changes in the XML file, for example, change the batch size value.
3. Import the updated profile into MDS by using the MDS import tool.

1.3.3 Changing the Profile Mode

You can use one of the following methods to change the profile mode property from CHANGELOG to REGULAR:

> **See Also:** "Mode of Reconciliation" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about changelog and regular reconciliation modes

- Change the value of the mode attribute in the profile, for example:

```
<generalconfig mode="REGULAR"
       createEntityUsingSPFlag="true"
       dateFormat="yyyy/MM/dd hh:mm:ss z"
       ownerMatchingRuleWhereClause="
       (UGP.ugp_ldap_guid=RA_SAMPLE_HIERARCHY.RECON_ROLE_GUID)"
       entitytype="RoleRole"
       version="1.0"
       trustedSrcFlag="false"
```

```
accountPostProcessingRequiredFlag="NOT_SET"
sequentialProcessingFlag="false"
batchSize="-1"
retryInterval="30"
maxRetryCount="5"
defaultProfileFlag="true"
name="sample-hierarchy"/>
```

- Change the attribute during event creation:

  The event creation API, introduced in Oracle Identity Manager 11*g* Release 1 (11.1.1), contains three parameters. The first two parameters are same as those used in previous create event APIs. The third parameter can have attributes such as dateFormat, changeType, eventFinished, and actionDate.

  You can use this API to set the changeType as follows:

```
public long createReconciliationEvent(String objName, Map<String, Object>
inputData, EventAttributes eventAttribs);
```

> **Note:** Using the API to set the changeType attribute overrides the value of the changeType attribute set in the profile.

1.4 Populating Data in the RECON_EXCEPTIONS Table

The RECON_EXCEPTIONS table in Oracle Identity Manager database is used to capture error messages generated during account reconciliation. This data is collected for the purpose of generating reports.

> **See Also:** "Account Reconciliation" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about account reconciliation

If a reconciliation match is found to a deleted user, then you must insert USER_DELETED in the REX_EXCEPTION column and the key of the deleted user in the USR_KEY column of the RECON_EXCEPTIONS table.

If no match is found, then insert USER_NOT_FOUND in the REX_EXCEPTION column.

If account match is found, then check if the account is already deprovisioned. Then insert into RECON_EXCEPTIONS table with the value `RESOURCE_DEPROVISIONED` in the REX_EXCEPTION column for the user who is to be provisioned.

To populate the RECON_EXEPTIONS table with exception data:

1. Fetch all the events with the change type != ('Modify' , 'Delete') and event status as ('Single User Match Found', 'Single Org Match Found').
2. Provision the resource object for the entities by performing the following:
   a. Collect the exception data from RECON_EXCEPTION DB table. To do so, perform any one of the following:

      Check if the value of the XL.EnableExceptionReports property is TRUE. If it is set to TRUE, then continue to the next step. Otherwise, do not collect the exception data.

      Select the obj_initial_recon_date in the obj table for the resource object being provisioned, and check if it is earlier than today's date. If an earlier

date is displayed, then continue to the next step. Otherwise, do not collect the exception data.

b. While provisioning the resource object to the user, check if the resource object has already been deprovisioned in Oracle Identity Manager:

If the resource object is already deprovisioned, then insert into RECON_EXCEPTIONS table the value `RESOURCE_DEPROVISIONED` in the REX_EXCEPTION column for the user who is to be provisioned.

If the resource object is not deprovisioned, then insert into RECON_EXCEPTIONS table the value `RESOURCE_NEVER_PROVISIONED` in the REX_EXCEPTION column for the user who is to be provisioned.

2

Managing Scheduled Tasks

In Oracle Identity Manager, it is often required to run jobs at specified times on a regular basis to manage various activities. Scheduler enables you to schedule jobs that automatically run predefined scheduled tasks at the specified time. This is illustrated by the following example:

To meet the security policies of an organization, employees may be required to change their product application password every 60 days. For this purpose, the system administrator has to ensure that an e-mail is sent to all employees whose passwords for the respective product applications have expired. One approach would be to identify the set of users whose passwords have expired and send e-mail to each employee manually. Alternatively, the system administrator can use a service such as scheduler. In Oracle Identity Manager, the system administrator can create a job named "Password Warning task" and schedule it or update the existing predefined task with the intended schedule.

Scheduler also enables you to create your own scheduled tasks that can be run by a job at a set time.

A **scheduled task** configure the metadata for a job, which is to be run, and the parameters required for execution of that task. This metadata is pre-defined for the predefined tasks. A new task can be added by the user, which will have the new metadata or the existing tasks can be updated to add/update the parameters for other configuration details. A **job** can be scheduled to run at the specified interval. You can create multiple jobs scheduled to run at different time intervals. A **job run** is a specific execution of a job. Each job run includes information such as the start time, stop time, exceptions and status of the execution.

This chapter discusses the following topics:

2.1 Configuring the oim-config.xml File

After you install Oracle Identity Manager, you can configure the scheduler settings by editing the child elements of the Scheduler element in the oim-config.xml file located in the following location:

OIM_HOME/metadata/db/oim-config.xml

Table 2–1 lists the default elements that you can configure within the Scheduler element in the oim-config.xml file.

Note: You can add new configurable child elements. For the information about new child elements, refer to the following URL:

`http://www.quartz-scheduler.org/`

Table 2–1 Child Elements of the Scheduler Element

Element Within Scheduler Element	Description
DSJndiURL	This element is used for configuring transactional data source in the application server, which is used by Quartz to establish the connection. Default value: jdbc/operationsDB
nonTxnDSJndiURL	This element is used for configuring non-transactional data source in the application server, which is used by Quartz to establish the connection. Default value: jdbc/oimJMSStoreDS
Clustered	Enter `true` if Oracle Identity Manager has been installed in a clustered environment. Otherwise, enter `false`. Default value: `true` **NOTE:** In a clustered environment, the clocks on all nodes of the cluster must be synchronized.
implementationClass	Enter the name of the Java class that implements scheduler. Default value: oracle.iam.scheduler.impl.quartz.QuartzSchedulerImpl
instanceID	Enter a unique string value in this element. This value represents a string that uniquely identifies an Oracle Identity Manager scheduler instance. **NOTE:** In a clustered environment, each node of the cluster must have a unique InstanceId. This can be achieved by entering a value of `AUTO` in the instanceId element.
startOnDeploy	Enter `false` if you do not want scheduler service to start automatically when Oracle Identity Manager is started. Otherwise, enter true. Default value: true
threadPoolSize	Enter an integer value in this element. This value represents the number of threads that must be used for running jobs. Default Value: 10

2.2 Starting and Stopping the Scheduler

The Scheduler Status page is an authenticated UI page that displays the current status of the scheduler. At any given instance, the scheduler can be in one of the following statuses:

- Started

  If the scheduler is in the started status, then jobs can be scheduled and jobs that have already been scheduled will continue to run at the scheduled time.

- Stopped

If the scheduler is in the stopped status, then all jobs that are currently running will be stopped and jobs that have been scheduled to run will not run but will be submitted for run as per schedule. As and when Scheduler Service will be up in future, all submitted jobs will be executed.

The Scheduler Status page also displays a detailed error message in the Last Error field, if any.

You can use the Scheduler Status page to either start, stop, or reinitialize the scheduler.

By default, the scheduler is in the started status after you install Oracle Identity Manager. However, if you want to stop scheduler for any reason and then restart it, then you must follow the procedure discussed in this section.

To start or stop the scheduler:

Note:

- You need to have Scheduler Admin role to start or stop the scheduler.
- In a clustered environment, you must perform this procedure on each node of the cluster.

---

1. Browse to the following URL by using a Web browser:

   http://*OIM_HOST*:*OIM_PORT*/SchedulerService-web/status

   In this URL, *OIM_HOST* represents the name of the computer hosting the Oracle WebLogic Application Server and *OIM_PORT* refers to the port on which the server is listening. The default port number for Oracle WebLogic Application Server is 7001.

2. Enter the User ID and password, and then click **OK.**

   The Scheduler Status page is displayed.

   Note: You may be automatically logged in to the scheduler service if you are working in a single sign-on environment.

   ---

3. Depending on the type of action that you want to perform, click one of the following:

   - **START:** Click this button to start the scheduler.
   - **STOP:** Click this button to stop the scheduler. This will stop all the running jobs and jobs that have been scheduled to run will run when the Scheduler Service is started again.
   - **REINIT**: Click this button to reinitialize the scheduler. Reinitializing the scheduler will restart the scheduler.

2.3 Scheduled Tasks

In Oracle Identity Manager, metadata is predefined for the default scheduled tasks. New tasks can be added by the user with new metadata, or the existing tasks can be updated to add or update the parameters or other configuration details.

For example, you can configure a reconciliation run using a scheduled task that checks for new information on target systems periodically and replicates the same in Oracle Identity Manager. Each scheduled task contains the following metadata information:

- Name of the scheduled task
- Name of the Java class that runs the scheduled task
- Description
- Retry
- (Optional) Parameters that the scheduled task accepts. Each parameter contains the following additional information:
  - Name
  - Data Type
  - Required/ Optional
  - Help Text
  - Encryption

This section discusses the following topics:

- Predefined Scheduled Tasks
- LDAP Scheduled Tasks
- Creating Custom Scheduled Tasks

2.3.1 Predefined Scheduled Tasks

This release of Oracle Identity Manager provides a set of predefined scheduled tasks that you can use while creating or working with jobs. Table 2–2 lists the predefined scheduled tasks.

Table 2–2 Predefined Scheduled Tasks

Scheduled Task	Description	User-Configurable Attributes
Password Expiration Task	This scheduled task sends e-mail to users whose password expiration date has passed at the time when the task was run and then updates the USR_PWD_EXPIRED flag on the user profile.	Email Definition Name: E-mail name that is sent to the user. The default value is "Password Expired".
Password Warning Task	This scheduled task sends e-mail to users whose password warning date had passed at the time when the task was run and then updates the USR_PWD_WARNED flag on the user profile.	Email Definition Name: E-mail name that is sent to the user. The default value is "Password Expiration Warning".
User Operations	This scheduled task performs the operation specified by the UserOperation attribute on the user account specified by the UserLogin attribute.	▪ UserLogin: User ID of the user account ▪ UserOperation: Operation that you want to perform on the user account. The value of this attribute can be ENABLE, DISABLE, or DELETE.
Attestation Grace Period Expiry Checker	This scheduled task delegates the attestation process after the grace period expires.	None

Table 2–2 (Cont.) Predefined Scheduled Tasks

Scheduled Task	Description	User-Configurable Attributes
Task Escalation	This scheduled task escalates pending tasks whose escalation time had elapsed at the time when the scheduled task was run.	None
Task Timed Retry	This scheduled task creates a retry task for rejected tasks whose retry time has elapsed and whose retry count was greater than zero.	None
Set User Deprovisioned Date	A deprovisioning date is defined when a user account is created. For users whose deprovisioning date had passed at the time when this schedule task was run, the task sets the deprovisioned date as the current date.	None
Disable/Delete User After End Date	An end date is defined when a user account is created. This scheduled task disables user accounts for which the end date had passed the current date at the time when the task is run.	None
Set User Provisioned Date	This scheduled task sets the provisioned date to the current date for users for whom all of the following conditions are true: ■ The provisioning date is in the past. ■ The deprovisioned date has not been set. ■ The deprovisioning date has not been reached or is NULL.	None
Enable User After Start Date	A start date is set when a user account is created. This scheduled task enables user accounts for which the start date has passed, and the user status is Disabled Until Start Date.	None
Scheduled Provisioning Task	When this scheduled task is run, it triggers scheduled request provisioning processes.	None
Remove Open Tasks	This scheduled task removes information about open tasks from the table that serves as the source for the list displayed in Oracle Identity Manager Administrative and User Console.	Day Limit Number of days for which information about an open task should be retained in the table before the information is deleted By default, this attribute is not specified and disabled. You must enable and configure the time.
Resubmit Reconciliation Event	This scheduled task resubmits reconciliation events whose status remains at Event Received for the time that you specify by using the window attribute.	window Number of hours for which the task has remained at the Event Received status
Issue Audit Messages Task	This scheduled task fetches audit message details from the aud_jms table and sends a single JMS message for a particular identifier and auditor entry in the aud_jms table. An MDB processes the corresponding audit message.	Max Records: Use this attribute to specify the maximum number of audit messages to be processed for a specified scheduled task run. The default value of this attribute is 400.
Initiate Attestation Processes	This scheduled task initiates a call to the Attestation Engine to run attestation processes that are scheduled to run at a time that has passed.	None

Table 2–2 (Cont.) Predefined Scheduled Tasks

Scheduled Task	Description	User-Configurable Attributes
Request Execution Scheduled Task	This is a periodic scheduled task searches for requests with status "Request Awaiting Completion" and moves requests forward to the next stage "Operation Initiated" if the effective date set during the request submission is prior or equal to the current date.	Job Periodic Settings: Use this attribute to specify the time interval for the scheduled task to be run. The default value is 6 hrs.
Automated Retry of Failed Async Task	This scheduled task retries Async Tasks (JMS Messages) that have failed. If the execution of the task succeeds, it is removed from the list of failed tasks. If it fails, the retry count is incremented. The maximum number of times a Failed Task is retried is determined by the 'maxRetries' defined for that task in async-messaging.xml.	None
Evaluate User Policies	This scheduled task re-evaluates the access policies.	Number of Threads: Use this attribute to specify the total number of threads that will process re-evaluation. The default value is 20. Batch Size: Use this attribute to fetch number of records from the database to be processed in one iteration. The default value is 500. Time Limit in mins: Use this attribute to specify time in minutes, after which the schedule task will stop. By default, this attribute is not specified and disabled. You must enable and configure the time.
Automatically Unlock User	This scheduled task automatically unlocks an user after the specified number of days.	None
Delayed Delete User	This scheduled task automatically deletes the user whose delete date is set as today.	None
Entitlement Assignments	This scheduled task populates Entitlement Assignment schema from child process form table whose field, Entitlement is marked as true.	RECORDS_TO_PROCESS_IN_BATCH: Number of records to process in a batch.
Entitlement List	This scheduled task populates Entitlement schema from lookup table whose child process form field, Entitlement is marked as true.	None
Entitlement Updates	This scheduled task populates Entitlement assignment table for a given user Entitlement Assignment Delta Table as & when Entitlements are add/update/delete for a User.	None
Get SOD Check Results Approval	This scheduled task gets back the result of SoD Evaluation from the SoD Server, for example, OAACG, SAP, and GRC for all requests waiting for SoD Check results. It reflects the SoDCheckResult and violation in appropriate dataset attributes. It will pick up all requests that are in "SoD check result pending" state and mark them as "SoD check completed".	None

Table 2–2 (Cont.) Predefined Scheduled Tasks

Scheduled Task	Description	User-Configurable Attributes
Get SOD Check Results Provisioning	This scheduled task gets back the result of SoD Evaluation from the SoD Server, for example, OAACG, SAP, and GRC for all pending SoDCheck provisioning tasks. It reflects the SoDCheckResult and violation in appropriate process form attributes.	None
Non Scheduled Batch Recon	This scheduled task tries to process all the events created by non scheduled task based connectors such as PeopleSoft. Such connector created events are in either Event Received State or Data Received State, they only get processed if the batch size specified by the set of events is reached or via this scheduled task. This task executes as per settings to pick up all the unprocessed non scheduled task based events and submits them to the reconciliation engine for processing.	None
Orchestration Process Cleanup Task	This scheduled task deletes all completed parent orchestration processes.	Batch Size: Use this attribute to specify the number of completed orchestration processes to be deleted in each iteration. Delete Just One Batch: Use this attribute to specify the value *true* or *false*. Only a single batch is deleted if the value is true. All the completed events are deleted batch at a time in a loop if the value is false.
Refresh Materialized View	The materialized view is used to generate reports related to reconciliation. This view needs to be updated periodically (at a specified interval, for instance, once a day). Therefore, this scheduled task was created to update the view on a periodic basis.	None
Resubmit Uninitiated Approval SODChecks	This scheduled task tries to initiate SoD Check for pending requests, which have SoDCheckStatus as "SoD check not initiated" or "SoD check completed with error". The pending requests are the ones for which SoD initiation failed in first try and are pending for some level of approval.	None
Resubmit Uninitiated Provisioning SODChecks	This scheduled task tries to initiate SoD Check by submitting a JMS message for all pending SoDCheck provisioning tasks. The SoD Check initiation may have failed because of SoD server being down at the time of entitlement add/update via direct provisioning.	None
Retry Failed Reconciliation Events	This scheduled task processes the failed reconciliation event for the users whose status is set as Failed.	None

Table 2–2 (Cont.) Predefined Scheduled Tasks

Scheduled Task	Description	User-Configurable Attributes
Run Future Dated Reconciliation Events	This scheduled task processes the current dated reconciliation event for the users whose status is set as Deferred.	None
Trigger User Provisioning	This scheduled task checks if provisioning date is in the past for each entry. If yes, it either sets the status to "Approved" which doesn't need a trigger, or to "Approved, Ready to Provision" which requires a trigger.	None
Job History Archival	This scheduled task is designed to archive/purge entries for Job History.	Archival Date: Use this attribute to specify date till which the records need to be archived/purged. Batch Size: Use this attribute to specify the size of a batch in which the records must be processed. Operation Type: Use this attribute to specify the operation type. This attribute can have two possible values, Archive and Purge. The default value is Archive.

2.3.2 LDAP Scheduled Tasks

This release of Oracle Identity Manager provides a set of LDAP scheduled tasks that you can use while creating or working with jobs. These schedule tasks are created only when Oracle Identity Manager is configured with LDAP synchronization. Table 2–3 lists the LDAP scheduled jobs.

> **See Also:** "Configuring the Integration with LDAP" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for detailed information about configuring the integration between Oracle Identity Manager and LDAP

Table 2–3 LDAP Scheduled Jobs

Scheduled Jobs	Description	User-Configurable Attributes
LDAP User Create and Update Reconciliation	This scheduled job reconciles user updates based on the change log from LDAP. The LDAP User Create and Update Reconciliation scheduled job cannot reconcile the User Defined Fields (UDFs). To enable this scheduled job to reconcilie UDFs, export the /db/LDAPUser and /db/RA_LDAPUSER.xml files from MDS, make required configuration changes in the files, and import them back to MDS. See "MDS Utilities and User Modifiable Metadata Files" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for information about importing and exporting MDS files. **Note:** While modifying the files, you must not specify any spaces when providing attribute names in the profile.	Last Change Number: Use this attribute to update the last change number of scheduled jobs with last changelog number value of Oracle Internet Directory. Batch Size: Use this attribute to fetch number of entries from the directory in each query. OIM User Type: User this attribute to specify the user type, for example, End-User or End-User Administrator. OIM User Organization Name: Use this attribute to specify Oracle Identity Manager organization in which reconciled users will be created. OIM Employee Type: Use this attribute to specify the value of employee type for users that are created through reconciliation.
LDAP User Delete Reconciliation	This scheduled job reconciles user deletes based on the change log from LDAP.	Last Change Number: Use this attribute to specify the last changelog identifier processed by this job. Batch Size: Use this attribute to fetch number of entries from the directory in each query.
LDAP Role Create and Update Reconciliation	This schedule job reconciles role creates or updates based on the change log from LDAP.	Last Change Number: Use this attribute to specify the last changelog identifier processed by this job. Batch Size: Use this attribute to fetch number of entries from the directory in each query.
LDAP Role Delete Reconciliation	This schedule job reconciles role deletes based on the change log from LDAP.	Last Change Number: Use this attribute to specify the last changelog identifier processed by this job. Batch Size: Use this attribute to fetch number of entries from the directory in each query.
LDAP Role Membership Reconciliation	This schedule job reconciles role membership based on the change log from LDAP.	Last Change Number: Use this attribute to specify the last changelog identifier processed by this job. Batch Size: Use this attribute to fetch number of entries from the directory in each query.

Table 2–3 (Cont.) LDAP Scheduled Jobs

Scheduled Jobs	Description	User-Configurable Attributes
LDAP Role Hierarchy Reconciliation	This schedule job reconciles role hierarchy based on the change log from LDAP.	Batch Size: Use this attribute to fetch number of entries from the directory in each query. Last Change Number: Use this attribute to specify the last changelog identifier processed by this job.
LDAP User Create and Update Full Reconciliation	This schedule job reconciles user creates or updates from LDAP, which includes all users under the search base that is defined in the Directory Server IT resource.	Batch Size: Use this attribute to fetch number of entries from the directory in each query. OIM Use Type: User this attribute to specify the user type, for example, End-User or End-User Administrator. OIM User Organization Name: Use this attribute to specify Oracle Identity Manager organization in which reconciled users will be created. OIM Employee Type: Use this attribute to specify the value of employee type for users that are created through reconciliation.
LDAP User Delete Full Reconciliation	This schedule job reconciles user deletes from LDAP. It detects the deleted users by comparing the users that exist in Oracle Identity Manager and LDAP.	Batch Size: Use this attribute to fetch number of entries from the directory in each query.
LDAP Role Create and Update Full Reconciliation	This schedule job reconciles role creates or updates from LDAP, which includes all roles under the search base that is defined in the Directory Server IT resource.	Batch Size: Use this attribute to fetch number of entries from the directory in each query.
LDAP Role Delete Full Reconciliation	This schedule job reconciles role deletes from LDAP. It detects the deleted roles by comparing the roles that exist in Oracle Identity Manager and LDAP.	Batch Size: Use this attribute to fetch number of entries from the directory in each query.
LDAP Role Membership Full Reconciliation	This schedule job reconciles role membership from LDAP. It detects the addition or deletion of role membership by comparing the entries existing in Oracle Identity Manager and LDAP.	Batch Size: Use this attribute to fetch number of entries from the directory in each query.
LDAP Role Hierarchy Full Reconciliation	This schedule job reconciles role hierarchy from LDAP. It detects the addition or deletion of role hierarchy by comparing the entries existing in Oracle Identity Manager and LDAP.	Batch Size: Use this attribute to fetch number of entries from the directory in each query.

2.3.3 Creating Custom Scheduled Tasks

Oracle Identity Manager provides you with the capability of creating your own scheduled tasks. You can create scheduled tasks according to your requirements if you choose not to use any of the predefined scheduled tasks listed in Table 2–2.

> **See Also:** "Creating Scheduled Task" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for detailed information about creating a scheduled task

To create a custom scheduled task:

1. Create the scheduled task xml file and seed it in MetaData Store (MDS).
2. Develop the schedule task class and package it in a Jar.
3. Upload the Jar by:
   - Using Plug-ins
   - Using Database

Using Plug-ins

You can upload the jar using the Plug-in Framework provided by Oracle Identity Manager.

To upload the jar using plug-ins:

1. Create the plugin.xml file.
2. Create the directory structure (plugin.zip) for the scheduled task.
3. Upload the created plugin.zip in the following location:

   `$OIM_HOME/plugins/`

Using Database

You can upload the jar in the database (DB) of Oracle Identity Manager.

To upload the jar using DB:

Upload the jar in DB using UploadJar utility. You can run this utility from the following location:

`$OIM_HOME/bin/`

> **See Also:** "Upload Jar Utility" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for information about running the UploadJar utility

2.4 Jobs

As discussed in one of the earlier chapters, a job is a task that can be scheduled to run at the specified interval. A job run is a specific execution of a job. Each job run includes information such as the start time, stop time, job status, exceptions and status of the execution.

This section discusses the following topics:

- Creating Jobs
- Searching Jobs
- Viewing Jobs
- Modifying Jobs
- Disabling and Enabling Jobs
- Deleting Jobs

2.4.1 Creating Jobs

Note: The procedure described in this section assumes that the XML file for the scheduled task, which contains the job description is available in the *OIM_HOME*/metadata/file directory.

To create a job:

1. Log in to Oracle Identity Administration with the appropriate credentials.
2. Click the **System Management** tab and then click **Scheduler**. Alternatively, you can click the "Search Scheduled Jobs" link on Welcome Screen.
3. On the left pane, from the **Actions** list, select **Create.** Alternatively, you can click the icon with the plus (+) sign beside the View list.
4. On the Create Job page, enter values in the following fields under the Job Information section:
   - **Job Name:** Enter a name for the job.
   - **Task:** Specify the name of the scheduled task that runs the job. Alternatively you can search and specify a scheduled task.

     To search and specify a scheduled task:

     a. Click the magnifying glass icon next to this field.

     b. In the Search and Select : Scheduled Task dialog box, specify a search criterion for the scheduled task and click the icon next to Search field.

        A list of all scheduled tasks that meet the search criterion is displayed.

     c. From this list, select the scheduled task that runs the job being created, and then click **Confirm.**
   - **Start Date:** Specify the date and time on which you want the job to run. To do this, select the date and time along with timezone from the date editor and click **Ok**. By default, the timezone is "(UTC-08:00) US Pacific Time".
   - **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 one of the following schedule types:
     - **Periodic:** Select this option if you want the job to be run at a time that you specify, on a repeating basis. If you select this option, then you must enter an integer value in the Run every field under the Job Periodic Settings section and select one of the following values:

       - mins

       - hrs

       - days
     - **Cron:** Select this option if you want the job to be run at a particular interval on a recurring basis. For example, you can create a job that must run at 8:00 A.M. every Monday through Friday or at 1:30 A.M. every last Friday of the month.

The recurrence of the job must be specified in the Cron Settings section. In the Recurring Interval field, you can select any of the following values:

- Daily

- Weekly

- Monthly on given dates

- Monthly on given weekdays

- Yearly

After selecting a value, you can enter an integer value in the Days between runs field.

– **Single:** Select this option if the job is to be run only once at the specified start date and time.

– **No pre-defined schedule:** This option specifies that no schedule is attached to the job you are creating, and therefore, it is not triggered automatically. As a result, the only option to trigger the job is by clicking **Save and Run Now**.

Note: For all the schedule types, if you want the job to be saved run immediately, then click **Save and Run Now.**

A message confirming that the job has been successfully created and triggered is displayed.

2.4.2 Searching Jobs

You can perform the following search operations to search for jobs in the Oracle Identity Administration:

- Performing a Simple Search for Jobs
- Performing an Advanced Search for Jobs

2.4.2.1 Performing a Simple Search for Jobs

To perform a simple search for jobs:

1. In the Welcome page of the Oracle Identity Administration, under System Management, click **Search Scheduled Jobs**. Alternatively, you can click the **System Management** tab, and then click **Scheduler.**
2. On the left pane, in the **Search** field, specify the search criterion for the job that you want to locate. You can also include wildcard characters in the search criteria.
3. Click the icon next to the Search field. A list of all jobs that meet the search criterion is displayed.

   The search results are displayed in a tabular format with the following columns:

   - Job Name: This column displays the name of the job. If you want to view the details of the job, then click its name in the column.
   - Status: This column displays the status of the Job. A job can be in any one of the following statuses:
     – RUNNING: The job is currently running.

- STOPPED: The job is currently not running. However, the job will run again at the date and time specified in the Next Scheduled Run field.
- INTERRUPT: The job is interrupted while running. This status may appear if admin server go down in between while job is running.
- FAILED: The Job was failed to execute due to some reasons.

2.4.2.2 Performing an Advanced Search for Jobs

To perform an advanced search for scheduler:

1. On the left pane of the Scheduler section, click **Advanced Search**. The Advanced Search: Scheduled Jobs page is displayed.
2. Select any one of the following options:
   - **All:** On selecting this option, the search is performed with the AND condition. This means that the search operation is successful only when all the search criteria specified are matched.
   - **Any:** On selecting this option, the search is performed with the OR condition. This means that the search operation is successful when any search criterion specified is matched.
3. In the Job Name field, enter the job name that you want to search. You can use wildcard characters in your search criteria. Select a search condition in the list adjacent to the Job Name field. The search conditions include Not Contains, Not Begins With, Not Equals, Equals, Ends With, Not Ends With, Contains, and Begins With.
4. For the Status field, select a search condition. Then select a status: **All**, **Running**, or **Stopped**.
5. In the Task Name field, enter the task name. You can use wildcard characters in your search criteria. Select a search condition in the list adjacent to the Task Name field.
6. Click **Search**. The list of jobs that match your search criteria are displayed in the search results table.

   Table 2–4 lists the columns of the search results table:

Table 2–4 Fields in the Search Results Table

Field	Description
Job Name	The name of the scheduled job
Task	The task associated with the job
Status	The status of the job, RUNNING, STOPPED, FAILED, or INTERRUPT
Schedule	The schedule or the time for the job to run
Last Run	The time when the job ran for the last time
Enable	The job is enabled or disabled

2.4.3 Viewing Jobs

To view the details of a job:

1. Search for the job whose details you want to view. See "Searching Jobs" on page 2-13 for information about how to search a job.

2. Click the job whose details you want to view in the Job Name column of the search results table.

The Job Details page is divided into the following sections:

- Job Information: This section displays the fields that provide information about the job. For example, Job Name, Task, Retries, and Start Date fields. If you want to modify the details of the job, then make the relevant change and click **Apply**. See "Modifying Jobs" on page 2-16 for more information about modifying jobs.
- Job Status: This section displays details of the status of the job in the following fields:
  - Current Status: This field displays the status of the job.
  - Last Run Start: This field displays the date and time of when the job started to run last.
  - Last Run End: This field displays the most recent date and time of when the job stopped running
  - Next Scheduled Run: This field specifies that no schedule is attached to the job you are creating and therefore the job is not triggered automatically. The only option to trigger the job in this case is performing "Run Now" .

    Note: No value is displayed in this field if the Schedule Type is No pre-defined schedule.

- Parameters: The parameter values specified are used at run-time while the job is being executed. The values need not be provided at the runtime, they can be there for each job and are used when the job is executed.
- Job History: This section displays a list of all job runs for the job in a table.

  Each row of the table displays the following information about the job:
  - Start Time: This column displays the date and time at which the job run started its run.
  - End Time: This column displays the time at which the job run ended its run.
  - Job Status: This column displays the status of the job.
  - Execution Status: This column displays the job execution status.

  You can reorder the display of columns in the table under the History section:

  a. From the View list, select **Reorder Columns.**

  b. In the Reorder Columns dialog box, select the column name that you want to move.

  c. Depending on the order in which you want to columns to appears, click the up or down arrows.

  To add or remove the columns displayed in the table under the History section:

  a. From the View list, select **Columns.**

  b. Depending on your requirement, select one of the following:

  - Show All

  - Start Time

- End Time

- Job Status

- Execution Status

c. Repeat Steps a and b for each column that you want to add or remove.

After viewing the details of the job, you can either modify, run, or stop the job. In addition, you can also enable or disable the job. Job Detail screen can be refreshed.

After you view the details of the job on the Job Details page, you can perform one of the following:

- If you want to modify the details of the job, then make the relevant change and click **Apply.** See "Modifying Jobs" on page 2-16 for more information about modifying jobs.
- If you want to run the job, then click **Run Now.**
- If the Disable button is enable, then it means that the job is currently enabled and you can disable the job by clicking **Disable.**
- If the Enable button is enable, then it means that the job is currently disabled and you and enable the job by clicking **Enable.**
- If you want to refresh a job detail screen, then click **Refresh**.
- If the Stop button is displayed, then it means that the job is currently running and you can stop the job by clicking **Stop**.

2.4.4 Modifying Jobs

To modify a job:

1. Search and view the details of the job that you want to modify. See "Viewing Jobs" on page 2-14 for information about viewing job details.

   Note: If you want to run the job, then click the job name in the first column of the search results table and then click **Run Now.** After you click **Run Now,** you need not perform the rest of the steps in this procedure. However, if you want to modify the job and then run it, then perform the next step and click **Run Now.**

2. On the Job Details page, you can modify all the details of the job, except for the Job Name and Task fields under the Job information section and the fields under the Job Status section. See Step 4 of "Creating Jobs" on page 2-12 for details about the fields that you want to modify.
3. Click **Apply** to commit the changes made on the Job Details page to the database.

   A message confirming that the job has been successfully modified is displayed.

2.4.5 Disabling and Enabling Jobs

In addition to creating and modifying jobs, you can disable a job that is currently enabled, and enable a job that has been disabled earlier. On the Job Details page:

- If the Enabled button is enable, then it means that the job is currently disabled and you can enable it by clicking **Enable.** A job that has been enabled will run only when one of the following is true on the Job Details page:

  - The date and time displayed in the **Start Date** field matches the current date and time.
  - The date and time displayed in the **Next Scheduled Run** field matches the current date and time.

- If the Disabled button is enable, then it means that the job is currently enabled and you can disable the job by clicking **Disable.** A job that has been disabled will not run even when the date and time on which the job has been scheduled to run matches the current date and time.

  To enable or disable a job:

  1. Search for the job that you want to enable or disable by performing the procedure described in "Searching Jobs" on page 2-13.
  2. On the left pane, in the search results table, right click on the job name and select **Enable** or **Disable**. Depending on whether you click **Enable** or **Disable**, a message indicating that the job has either been successfully enabled or disabled is displayed.
  3. Click **OK** to close the dialog box.

2.4.6 Starting and Stopping Jobs

In addition to scheduling jobs to run automatically at the specified time, you can manually start or stop a job at any given time. For example, you create and schedule a job that runs every Friday. However, if you want to run the job on any day other than Friday, then you must run the job manually.

To start or stop a job:

1. Search for the job that you want to start or stop by performing the procedure described in "Searching Jobs" on page 2-13.
2. On the left pane, in the search results table, click the job name of the job that you want to start or stop.

   Note: By default, the status of all jobs is STOPPED unless a job is running.

3. If you want to start a job, then from the Actions list, click **Run Now.**

   A dialog box prompting you to confirm if you want to run the job is displayed.
4. If you want to stop a job, then from the Action list, click **Stop.**

   A dialog box prompting you to confirm if you want to stop the job is displayed.
5. Click **OK.**

2.4.7 Deleting Jobs

To delete a job:

1. Search for the job that you want to delete by performing the procedure described in "Searching Jobs" on page 2-13.
2. On the left pane, in the search results table, click the job name of the job that you want to delete.

3. From the Actions list, click **Delete.** Alternatively, you can click the Delete icon next to the icon with the plus (+) sign.

   A dialog box prompting you to confirm if you want to delete the job is displayed.

4. Click **OK.** A message indicating that the job has been deleted successfully is displayed.

3

Managing Notification Templates

Information about events occurring in Oracle Identity Manager are required to be sent to various users, such as requesters, beneficiaries, or administrators. This information about events is sent by using the notification service in the form of notification e-mail messages. The notification service allows you to perform all notification-related operations in Oracle Identity Manager.

An event is an operation that occurs in Oracle Identity Manager, such as user creation, request initiation, or any custom event created by the user. The events are generated as part of business operations or generation of errors. Event definition is the metadata that describes the event. Metadata for events are provided by identifying all event types supported by a functional component. For example, as a part of the scheduler component, metadata can be defined for schedule job execution failed and shutting down of the scheduler. Every time a job fails or the scheduler is shut down, the events are raised.

The data available in the event is used to create the content of the notification. The different parameters defined for an event help the system to select the appropriate notification template. The different parameters that are defined for an event help the system to decide what all event variables can be made available at the template design time.

A notification template is used to send notifications. These templates contain variables that refer to available data to provide more context to the notifications. The channel through which a notification is sent is known as the notification provider. For this release, only e-mail is available as the notification provider. At the backend, the notification engine is responsible for generating the notification, and utilizing the notification provider to send the notification.

> **Note:** An IT resource of name `Email Server` and type `Mail Server` must be configured for sending notifications. The email server settings gets cached. If there is a change in detail of an already configured email server, then you must restart the application server or delete the cache, as required.

Oracle Identity Manager provides a set of default notification templates, as shown in Table 3–1.

Table 3–1 Default Notification Templates

Notification Template	Description
Bulk Request Creation	Template to notify during a bulk parent request creation

Table 3–1 (Cont.) Default Notification Templates

Notification Template	Description
Create User Self Service Notification	Template to notify after a new user is created
End Date	Template to notify the manager when end date of the reportee expires
Generated Password Notification	Template to notify after a password is generated by Oracle Identity Manager
Request Creation	Template to notify during a request creation
Request Identity Creation	Template to notify during a Create User request
Request Status Change	Template to notify during a request status change
Reset Password	Template to notify after password has been reset
User Deleted	Template to notify the manager when the user account of the reportee is deleted as a result of expired end date
Add Proxy Notification	Template to notify after a proxy has been added for a user

Notification templates are described in the following sections:

- Defining Event Metadata
- Creating a Notification Template
- Searching for a Notification Template
- Modifying a Notification Template
- Deleting a Notification Template
- Adding and Removing Locales from a Notification Template

3.1 Defining Event Metadata

Corresponding to each event, you must create an XML file that has the specific schema defined by the notification engine. Compliant to that schema (.xsd file), an XML file is created that defines how an event looks like. When the event is defined, you can configure a notification template for that event.

An event file must be compliant with the schema defined by the notification engine, which is NotificationEvent.xsd. The event file contains basic information about the event.

> **Note:** The NotificationEvent.xsd file is in the iam\iam-product\features\notification\metadata directory in the MDS.

The following is a sample event XML file:

```
<?xml version="1.0" encoding="UTF-8"?>
<Events xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../../../metadata/NotificationEvent.xsd">
  <EventType name="User Created">
    <StaticData>
      <Attribute DataType="X2-Entity" EntityName="User" Name="Granted User"/>
      <Attribute DataType="X2-Entity" EntityName="User" Name="Grantee User"/>
```

```
      <Attribute DataType="91-Entity" EntityName="User Group" Name="User Grp"/>
    </StaticData>
      <Resolver class="oracle.iam.notification.DemoResolver">
      <Param DataType="91-Entity" EntityName="Resource" Name="ResourceInfo"/>
    </Resolver>
  </EventType>
</Events>
```

In the sample XML file:

- The EventType name element is the name of the event.
- The StaticData element lists a set of parameters that lets the user add parameters that are not data dependent. In other words, this element defines the static data to be displayed when notification is to be configured. An example of static data is the user entity, which is not dependent on any other data and has the same set of attributes for all event instances and notification templates.
- The Param DataType element lists a set of parameters lets the user add parameters that are data dependent. An example of the data dependent or a dynamic entity is a resource object that the user can select at run time. A notification template is to be configured for the resource object. Corresponding to the resource object field, a lookup is displayed on the UI. When a user selects the event, for example End Date Notification, the call goes to the Resolver class provided to fetch the fields that are displayed in the Available Data list, from which user can select the attribute to be used on the template.

  Note: Available data is the list of attributes that can be embedded as a token in the template. These tokens are replaced by the value passed by the resolver class at run time. Available data is displayed in a list.

  Selected data is a single attribute that helps user to copy and paste the attribute name. Selected data is the same attribute name as selected in the Available Data list.

  The dynamic entities supported for lookup are user, resource, and organization. These entity names must be specified in the Param DataType element.
- The Resolver class must be defined for each notification. It defines what parameters are available in the notification creation screen and how those parameters are replaced when the notification is to be sent. In other words, the resolver class resolves the data dynamically at run time and displays the attributes in the UI.

  See Also: "Notification Tab" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for more information about the resolver class and the methods supported by the resolver class

The event XML file is uploaded into MDS by using the MDS import and export utility. Notification service reads the XML files from MDS.

See Also: "MDS Utilities and User Modifiable Metadata Files" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for information about using the MDS import and export utilities

3.2 Creating a Notification Template

> **Note:** Corresponding to each event that happens, you have to configure an XML file. The XML file defines the behavior of each event. Only after an XML is configured for an event, you can create a notification template for that event.
>
> For information about creating the event XML file, see "Defining Event Metadata" on page 3-2.

To create a notification template:

1. Log in to the Oracle Identity Administration.
2. Click the **System Management** tab and then click the **Notification** tab.
3. From the Actions list on the left pane, select **Create.**
4. On the Create page, enter values for the following fields under the Template Information section:
   - **Template Name:** Enter the template name in this field.
   - **Description Text:** Enter a brief description of the template in this field.

   > **Note:** The Description Text field cannot be translated and is available only in English.

5. Under the Event Details section, perform the following:
   - From the Available Event list, select the event for which the notification template is to be created from a list of available events. Depending on your selection, other fields are displayed in the Event Details section.
   - In the Resource field, select a resource from the lookup. This is the dynamic data defined by the Param DataType element in the XML definition. For more information about this element, see "Defining Event Metadata" on page 3-2.
6. Under the Locale Information section, enter values in the following fields:

   > **Note:** The Default Locale information is stored in the PTY table and is fetched from there.

   - To specify a form of encoding, select either UTF-8 or ASCII.
   - In the **Message Subject** field, enter a subject for the notification.
   - From the **Type** options, select the data type in which you want to send the message. You can choose between HTML and Text/Plain.
   - In the **Short Message** field, enter a gist of the message in very few words.
   - In the **Long Message** field, enter the message that will be sent as the notification.
7. After you have entered the required values in all the fields, click **Save.**
8. A message is displayed confirming the creation of the notification template. Click **OK.**

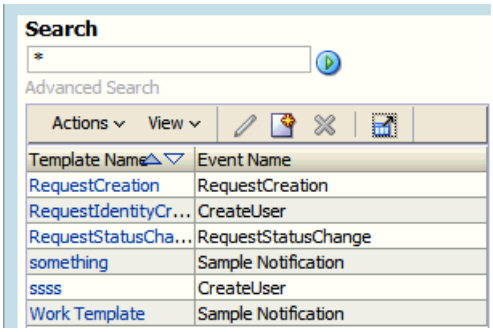
3.3 Searching for a Notification Template

You can perform a simple search or an advanced search for a notification template by using Advanced Administration.

To perform a simple search for a notification template:

1. Log in to the Oracle Identity Administration.
2. Click the **System Management** tab and then click the **Notification** tab.
3. Click the icon next to the **Search** field. All the existing notification templates are displayed on the left pane, as shown in Figure 3–1:

Figure 3–1 Notification Search Result

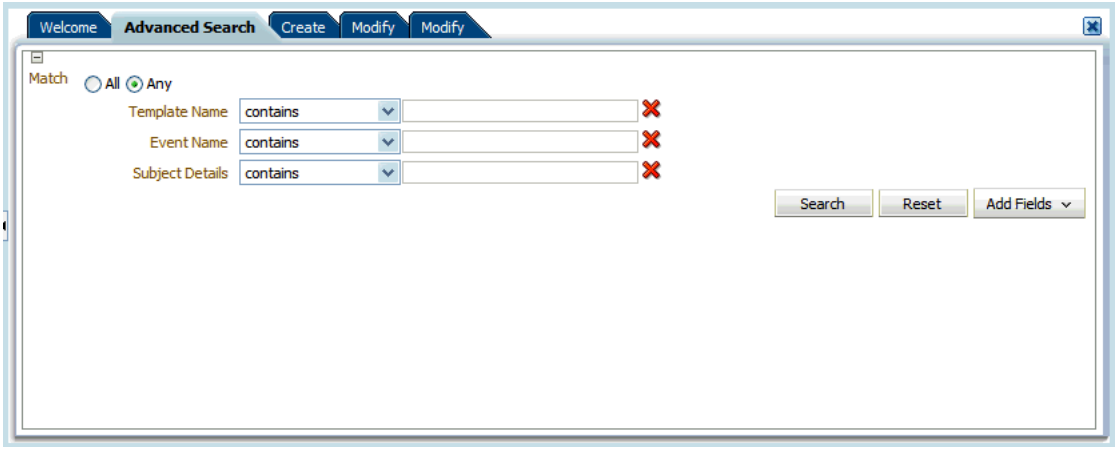


4. Select the template that you want to view. The details of the selected notification template are displayed on the right pane.

To perform an advanced search for a notification template:

1. In the left pane of the Oracle Identity Administration, click **Advanced Search**. The Advanced Search page is displayed, as shown in Figure 3–2:

Figure 3–2 The Advanced Search Page

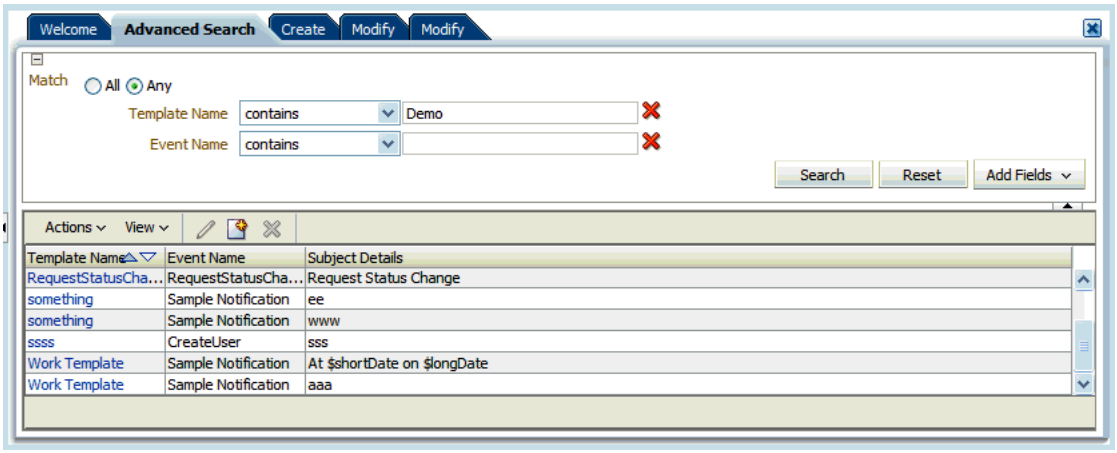


2. Select one of the following matching options:
   - **All:** On selecting this option, the search is performed with the AND condition. This means that the search operation is successful based on Search field with any input from the user. Search field with no input from the user is not considered.

- **Any:** On selecting this option, the search is performed with the OR condition. This means that the search operation is successful when any search criterion specified is matched.

3. Specify the search criteria in the Template Name, Event Name, and Subject Details fields. You can remove any of these fields that you do not want to include in the search by clicking the icon next to it. You can add a field that you want to include in the search by clicking **Add Fields**, and then selecting the field name from the list.
4. Click **Search**. The search results table is displayed with details about template names, event names, and subject details, as shown in Figure 3–3:

Figure 3–3 Advanced Search Results

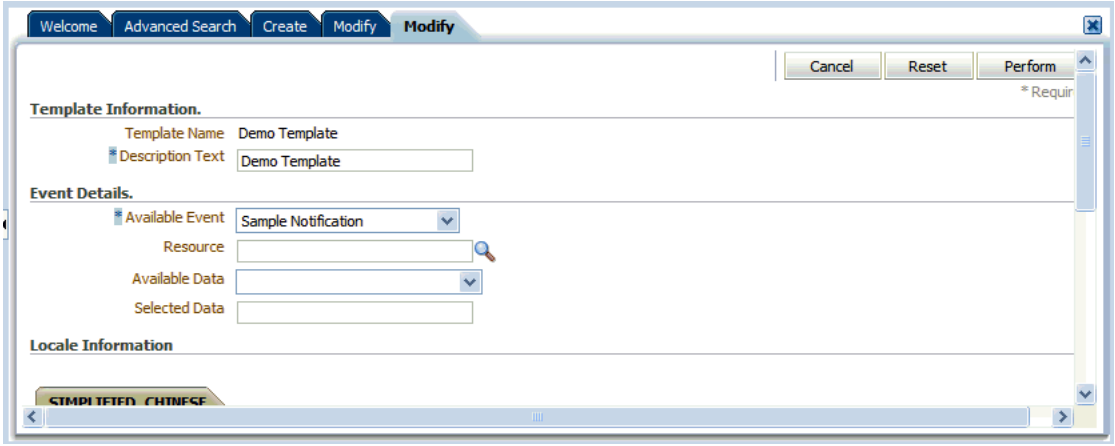


3.4 Modifying a Notification Template

To modify a notification template:

1. Log in to the Oracle Identity Administration.
2. Click the **System Management** tab and then click the **Notification** tab.
3. Click the icon next to the **Search** field. All the existing notification templates are displayed on the left pane. Select the template that you want to modify. Figure 3–4 shows the details of a notification template.

Figure 3–4 Notification Template Modification



4. Change the values that you want to and click **Save.**
5. A message is displayed confirming the modification of the notification template. Click **OK.**

3.5 Deleting a Notification Template

To delete a notification template:

1. Log in to the Oracle Identity Administration.
2. Click the **System Management** tab and then click the **Notification** tab.
3. Click the icon next to the **Search** field. All the existing notification templates are displayed on the left pane. Select the template that you want to delete.
4. From the Actions list, click **Delete.** A message is displayed prompting you to confirm the delete the operation. Click **OK.** A message is displayed confirming the delete operation.

3.6 Adding and Removing Locales from a Notification Template

To add locales to a notification template:

1. Log in to the Oracle Identity Administration.
2. Click the **System Management** tab and then click the **Notification** tab.
3. Click the icon next to the **Search** field. All the existing notification templates are displayed on the left pane. Select the template that you want to add a locale to.
4. From the Actions list, select **Add Locale.** The Add Locale page is displayed. In the Locale Name field, click the icon next to the Locale Name field to select a locale from a list. After selecting the locale, click **Confirm.** Click **Next.** The Locale Information page is displayed and the locale that you added is displayed as a tab in the page.
5. In the Locale Information section, specify values for all the fields as mentioned in step 6 of "Creating a Notification Template" on page 3-4 and then click **Save**. The locale is added to the template.

Note: Notification can be sent in all the locales that are added to the notification template. A user receives notification in the same locale specified in the user preferences. If a locale is not specified in the user preferences, then the notification is sent in the default locale. The default locale is to be specified in the PTY table in Oracle Identity Manager database at the time of installation.

To remove locales from a notification template:

1. In the left pane of the Oracle Identity Administration, select the template from the search results table, and click **Remove Locale** from the Actions list. Alternatively, you can right-click the template, and select Remove Locale.
2. On the Remove Locale page, click the icon next to the Locale Name field to select a locale from a list . Remember, you can remove a locale from a template only if that template contains multiple locales. You cannot remove a locale if it's the only one associated with the template. Click **Save.**
3. A message is displayed confirming the removal of the locale. Click **OK.**

Note: You must not remove default locale to ensure that a notification is sent every time when there is no user preferred locale is set or when notification template does not contain a locale template matching to user preferred locale.

4

Administering System Properties

The system configuration service enables you to manage system properties used by Oracle Identity Manager. This service allows you to create, modify, delete, or search existing system properties depending on their roles.

System properties define the characteristics that control the behavior of Oracle Identity Manager. You can define the functionality of consoles such as the Oracle Identity Administration and Oracle Identity Manager Self Service by using system properties. For example, you can define the number of consecutive attempts the user can make to login to Oracle Identity Manager unsuccessfully before Oracle Identity Manager locks the user account. In other words, a system property is an entity by using which you can control the configuration of Oracle Identity Manager.

This chapter discusses the following topics:

4.1 System Properties in Oracle Identity Manager

Table 4–1 lists and describes the default system properties in Oracle Identity Manager.

Table 4–1 Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Are challenge questions disabled in OIM	Determines if challenge questions are enabled or disabled when a user logs in to Oracle Identity Manager for the first time. When value is False, challenge questions are enabled. When value is True, challenge questions are disabled. This property is primarily used in the context of Oracle Adaptive Access Manager (OAAM) configuration. When the value is TRUE, the challenge questions are handled by OAAM.	OIM.DisableChallengeQuestions	FALSE
Compiler Path for Connectors	Specifies the Java home depending on the application server. **Note:** If the path of the JDK directory is not included in the System Path variable, then you must set the path of the JDK directory in the XL.CompilerPath system property. If this is not done, then an error is encountered during the adapter compilation stage of the process performed when you import an XML file by using the Deployment Manager.	XL.CompilerPath	
Default Date Format	When creating reconciliation events by calling the APIs and date format is not passed as one of the arguments to the API, Oracle Identity Manager assumes that all the date field values are specified in Default Date Format. If no value is set for this system property, Oracle Identity Manager assumes the format to be yyyy/mm/dd hh:mm:ss z.	XL.DefaultDateFormat	yyyy/mm/dd hh:mm:ss z
Default policy for username generation	Determines the username policy to be picked up while generation of username.	XL.DefaultUserNamePolicyImpl	oracle.iam.identity.usermgmt.impl.plugins.DefaultComboPolicy
Default user name domain	This property is used by the DefaultComboPolicy to generate a user name in e-mail format.	XL.UserNameDomain	oracle.com

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Direct Provisioning vs. Request for Access Policy Conflicts	If multiple access policies are evaluated to be true for a user that requires the same resource to be provisioned and some of the policies are defined to provision resource with approvals and some without approval, and if the Direct Provisioning vs. Request for Access Policy Conflicts property is set to FALSE, then Oracle Identity Manager creates a request for provisioning the corresponding resource. If there are no conflicts, then resources are provisioned based on what is defined on the access policy.	XL.DirectProvision	TRUE
Does user have to provide challenge information during registration	If the value is TRUE, then users will have to provide challenge information during registration.	PCQ.PROVIDE_DURING_SELFREG	TRUE
Duplicate challenge responses allowed	This property is used to indicate whether or not duplicate challenge responses are allowed.	XL.IsDupResponsesAllowed	FALSE
Email Server	Name of the e-mail server. **Note**: After modifying the Email Server system property value, you must restart the server for the change to take effect.	XL.MailServer	Email Server
Enable exception reports	This property is used to enable the exception reporting feature. Exception reporting is enabled only if the value is set to TRUE.	XL.EnableExceptionReports	FALSE
Enable disabled resource instances when a user is enabled	If the value is TRUE, then the disabled resource instances are enabled when a user is enabled.	XL.EnableDisabledResources	TRUE

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Flag for new permissioning model	This system property determines the data object permission model for inserting, updating, and deleting records in the Oracle Identity Manager database. Before inserting, updating, and deleting records into a database table, Oracle Identity Manager checks the roles assigned to the user who wants to insert, update, or delete records. The roles have data objects assigned to them along with details of permissions to insert, update, or delete a record. For a user to insert, update, or delete records into the table, the user must have permissions for the all the roles assigned to him on that data object. If the user does not have insert, update, or delete permission on any one role, then the user is not allowed to insert, update, or delete records in the table corresponding to the data object. This applies when the value of this property is set to FALSE. When the value is set to TRUE, the user must have insert, update, and delete permissions for any one of the roles assigned to the user on a particular data object. If any one permission is available to the user for a role, then the user can insert, update, or delete records in the table corresponding to the data object.	XL.NewPermissionModel	FALSE
Force Password Change at First Login	If the value is TRUE, then the user is forced to change the password when the user logs in for the first time.	XL.ForcePasswordChangeAtFirstLogin	TRUE
Force to set questions at startup	When the user logs into the Web Application for the first time, he/she must set the default questions for resetting his/her password.	PCQ.FORCE_SET_QUES	False
Is Self-Registration Allowed	If the value is TRUE, then the users are allowed to self-register.	XL.SelfRegistrationAllowed	TRUE
LDAP Reservation Plugin	This property determines the LDAP reservation plugin implementation to be picked up for reservation of user attributes.	XL.LDAPReservationPluginImpl	oracle.iam.identity.usermgmt.impl.plugins.reservation.ReservationInOID

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Maximum Number of Login Attempts	Determines how many consecutive times the user can attempt to login to Oracle Identity Manager unsuccessfully before Oracle Identity Manager locks the user account. **Note**: If the user account is locked, then the user can unlock it by resetting the "challenge" questions associated with resetting the password.	XL.MaxLoginAttempts	10
Maximum Number of Password Reset Attempts	Determines how many consecutive times the user can attempt to reset the password unsuccessfully before Oracle Identity Manager locks the user account. **Important**: When the user account is locked, the user cannot unlock it. If this occurs, then contact the system administrator.	XL.MaxPasswordResetAttempts	3
Minimum length of challenge response	This property is used to set the minimum length of answers to challenge questions.	XL.ResponseMinLength	0
Number of Correct Answers	This value represents how many questions the user must answer correctly to reset user password.	PCQ.NO_OF_CORRECT_ANSWERS	3
Number of Questions	Sets the number of questions that must be completed by a user who is using the Web Application to reset the user's password.	PCQ.NO_OF_QUES	3 **Note:** The value set for PCQ.NO_OF_QUES must not be less than the value set for PCQ.NO_OF_CORRECT_ANSWERS.
Organization Delete/Disable Action	If this property is set to TRUE, then users can disable/delete the organization even if the organization contains users and suborganizations. If this property is FALSE, then users cannot disable/delete the organization if the organization contains users and suborganizations. The default value is FALSE.	ORG.DisableDeleteActionEnabled	FALSE
Organization Process Inheritance	If a resource is added to an organization as permitted resource, then by setting this property to TRUE, the same resource is automatically added as the permitted resource for suborganizations.	XL.OrganizationProcessInherit	TRUE

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Organization Process Restriction	This property is for internal use by Oracle Identity Manager. You must not use this property.	XL.OrganizationProcessRestrict	FALSE
Organization Self-Serviceable	Determines whether the default value for a process is self-serviceable and if it is set or not. This is used to determine which resources can be self requested. This is same as selecting the option from Oracle Identity Manager Design Console. . The only difference is that from here it is allowed for a particular organization.	ORG.SELF_SERVICEABLE_DEFAULT	FALSE
Pending Cancelled Tasks	If this property is set to TRUE and tasks are configured to allow cancellation while they are pending, then these tasks are moved to Pending Cancelled (PX) status if the corresponding process instance is cancelled. If the property is set to FALSE, then tasks are moved to Cancelled (X) status when corresponding process instance is cancelled. Note that process instances are called by Oracle Identity Manager when the corresponding resource instances are revoked.	XL.PendingCancelled	true
Period to Delay User Delete	This property is used to specify the time period before deleting a user. When this property is set and a user is deleted, the user's state is changed to disabled and "automatically delete on date" is set to current date plus the delay period.	XL.UserDeleteDelayPeriod	0
Property dictates whether database name will be displayed	If the value is TRUE, then the database name is displayed.	XL.TOOLBAR_DBNAME_DISPLAY	TRUE
Property to indicate day limit set for pending approvals	Used prior to implementation of the Separation of active/non-active task feature to specify the duration for which the pending approval tasks would be fetched. Used at the API level to get the Pending approval related counters.	XL.OpenTask.DayLimit	30

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Property to indicate the duration in months of open tasks and pending approvals	**Note:** Do not use this property. It is retained in this release for internal use only. It will be removed in a future release of Oracle Identity Manager.	XL.OpenTasksPendingApprovalsDuration	3
Property to indicate whether the auditing engine should send a JMS message	When the value of this property is set to True and the XL.UserProfileAuditDataCollection property is set to an audit data collection level, then the account reconciliation performs the matching in the database layer at a batch-level and performs the event action by using the provisioning APIs. This in turn triggers the audit event handlers for account reconciliation. For information about account reconciliation, see "Account Reconciliation" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager.* **Note:** This property is for internal use by Oracle Identity Manager. You must not use this property.	XL.SendAuditJMSMessage	false
Proxy User Email Notification	The corresponding PTY_VALUE is the e-mail definition name that is sent when a proxy user is created. User gets a notification e-mail when the user is made proxy for some other user.	XL.ProxyNotificationTemplate	Notify Proxy User
Recon Batch Size	This property is used to specify the batch size for reconciliation. You can specify 0 as the value for this to indicate that the reconciliation will not be performed in batches. You must restart Oracle Identity Manager after setting this property.	OIM.ReconBatchSize	500
Record Read Limit	Sets the maximum number of records that can be displayed in a query result set.	XL.READ_LIMIT	500
Request Notification Level	This property indicates whether or not notification is sent to the requester and beneficiary when a request is created or the request status is changed. When the value of this property is 0, then the notification feature is disabled. When the value is 1, then the notification feature is enabled.	RequestNotificationLevel	0

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Reset with generated password	This property is used to reset the password of the user with generated password.	XL.ResetWithGeneratedPwd	TRUE
Search Stop Count	This property determines the maximum number of records that are displayed in the advanced search result. If the search criteria specified returns more number of records than that value of this property, then the number of records displayed is limited to this value. In addition, a warning is displayed stating that the results exceed maximum counts and you must refine your search with additional attributes.	XL.IDADMIN_STOP_COUNT	300
Shows tasks assigned to group users with least load only	If the value is TRUE, then the tasks are assigned to group users with least load only when the assignment type is Group User With Least Load, and so on.	XL.ShowTaskAssignedToGroupUserOnly	FALSE
Specifies the LDAP container mapper plug-in to be used	When Oracle Identity Manager is installed with LDAP synchronization enabled, this plug-in determines in which container users and roles are to be created. Value of this system property indicates the default Oracle Identity Manager plug-in name used for computing the container values. If the default plug-in does not meet the requirement, then you can define your own plug-in to determine the container and specify the name of the plug-in in this system property. **Note:** For information about this plug-in, see "Customizing Operations in Oracle Identity Manager" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.	LDAPContainerMapperPlugin	oracle.iam.ldapsync.impl.DefaultLDAPContainerMapper
Specifies the request engine to be used	This property is used to specify the request engine to be used for generating requests.	XL.RequestEngine	1
URL for challenge questions modification	This property is used in combination with the property OIM.DisableChallengeQuestions. The value of this property is the URL within OAAM that handles the challenge questions. For example: http://*OAAM_HOST*:*OAAM_PORT*/*OAAM_SERVER*/userPreferences.do?showView=registerQuestions	OIM.ChallengeQuestionsModificationURL	NONE

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
URL for change password	This property is used in combination with the property OIM.DisableChallengeQuestions. The value of this property is the URL within OAAM that handles the change password functionality. For example: http://*OAAM_HOST*:*OAAM_PORT*/*OAAM_SERVER*/userPreferences.do?showView=changePassword	OIM.ChangePasswordURL	NONE
URL for forgot password	This property is used in combination with the property OIM.DisableChallengeQuestions. The value of this property is the URL within OAAM that handles the forgot password functionality. For example: http://*OAAM_HOST*:*OAAM_PORT*/*OAAM_SERVER*/forgotPassword.do	OIM.ForgtoPasswordURL	NONE
Unlock Account Automatically After Time Period	This property is used to automatically unlock user accounts after the specified time period.	XL.UnlockAfter	FALSE
Use Row Restriction	**Note:** This property is for internal use by Oracle Identity Manager. You must not use this property.	XL.UseRowRestriction	FALSE
Use of Default Questions	For customers who have customized their UI to allow end-users to set their own challenge questions, this property determines whether the user must select challenge questions from a predefined list in the Web Application, or if users are required to provide their own questions. **Note**: Functionality that allows end-users to set their own challenge questions is not supported in the standard out-of-the-box user interface.	PCQ.USE_DEF_QUES	TRUE

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Use semicolon as delimiter in API parameters	This property is used to specify whether or not semicolon should be used as a delimiter to the API input parameter values. Some APIs accepted string input values that are separated by semicolon. This has been changed to use a vertical bar "|" instead. To keep backward compatibility, this new property can be used to go back to using semicolons. The default value is FALSE signifying the usage of "|". When set to TRUE, the input for those APIs are accepted with semicolon as separator.	XL.UseSemiColonAsDelimiter	FALSE
User Attribute Reservation Enabled	This property is used to enable user attribute reservation.	XL.IsUsrAttribReservEnabled	TRUE
User Id reuse property	Determines whether a deleted user account can be reused. To reuse a deleted user account, assign this property a value of TRUE and drop the unique index for the USR_LOGIN column in the USR table and create a nonunique index. To prevent a user account from being reused, assign this property a value of FALSE.	XL.UserIDReuse	FALSE
User Language	The user.language value is configured during installation for Locale handling at server side.	user.language	en
User Region	The user.region value is configured during installation for Locale handling at server side.	user.region	US
User Variant	The user.variant value is configured during installation for locale handling at server side.	user.variant	

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
User profile audit data collection level	This property controls the user profile data that is collected for audit purpose when an operation is performed on the user, such as creation, modification, or deletion of a user, role grants or revokes, and resource provisioning or deprovisioning. Depending upon the property value, such as Resource Form or None, the data is populated in the UPA table. The audit levels are specified as values of this property. The supported levels are: ■ **Process Task:** Audits the entire user profile snapshot together with the resource lifecycle process. ■ **Resource Form:** Audits user record, role membership, resource provisioned, and any form data associated to the resource. ■ **Resource:** Audits the user record, role membership, and resource provisioning. ■ **Membership:** Only audits the user record and role membership. ■ **Core:** Only audits the user record. ■ **None:** No audit is stored.	XL.UserProfileAuditDataCollection	Resource Form
XL.SoDCheckRequired	This property indicates whether or not SoD check is required.	XL.SoDCheckRequired	FALSE

Table 4–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Description	Keyword	Default Value
Xellerate User resource provision mode	This property determines whether provisioning of the Xellerate User resource to the user's organization occurs in the database layer through stored procedure, or in the Java layer via Event Handlers. **Note:** See *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for information about Event Handlers. This property has the following allowed values: ■ **DB:** Provisioning of the Xellerate User resource to the user's organization occurs in the database layer through stored procedure. This in turn does not trigger any further process. Therefore, custom tasks associated with the Xellerate User provisioning process that is associated with the Xellerate User resource does take place. ■ **Java:** Provisioning of the Xellerate User resource to the user's organization occurs in the database layer via Event Handlers. Custom tasks associated with the Xellerate User provisioning process that is associated with the Xellerate User resource takes place. This is applicable to the upgrade scenario, where you have your own tasks associated with provisioning processes in earlier releases of Oracle Identity Manager, and you want them to run even after 11*g* upgrade. In such scenario, set the value of this property value to JAVA.	XLUserResource.ProvisionMode	DB

Table 4–2 lists the system properties you can add to the PTY table, and then use the properties to change some of the default settings in Oracle Identity Manager:

Table 4–2 Nondefault System Properties

Property Name	Description	Keyword	Sample Value
OIM Database Query Retry Attempts	Number of times SQL queries to be retried for handling Oracle RAC failures. In the absence of this property in the PTY table, SQL queries for handling Oracle RAC failures are retried three times by default.	OIM.DBQueryRetryAttempts	5
OIM Database Query Retry Interval	Time in seconds after which each SQL retry takes place for Oracle RAC failures. In the absence of the property in the PTY table, SQL query occurs after every 7 seconds by default.	OIM.DBQueryRetryInterval	10 seconds
JDBC Connection Retry Attempts	Number of times Oracle Identity Manager retries to get a connection when the JDBC connection fails. In the absence of this property in the PTY table, the JDBC connection is retried three times by default.	OIM.JDBCConnectionRetryAttempts	5 When the value is 0, it means no retry.
JDBC Connection Retry Interval	Time in seconds between each JDBC connection retry. In the absence of this property in the PTY table, each JDBC connection retry occurs at an interval of 7 seconds.	OIM.JDBCConnectionRetryInterval	10 seconds
GTC Auto Import	Based on the value of this property, the DM xml that is generated while GTC creation can be saved to a directory. The default value of this property is true. When the value of this property is set to "False", then while creating GTC, the DM xml (the xml that GTC creates and imports using Deployment Manager internally while GTC creation) created by the GTC framework is stored in the following directory: O*IM_HOME*/GTC/XMLOutput The naming convention followed for the DM xml is: *GTCNAME_CURRENTDATE_TIMESTAMP* created using date format "yyyy-MM-dd-HH-mm-ss".xml For example: TRUSTEDCSV_2009-02-05-22-41-11.xml	XL.GTCAutoImport	False

4.2 Creating and Managing System Properties

This section discusses the following topics:

- Creating System Properties
- Purging Cache
- Searching for System Properties
- Modifying System Properties
- Deleting System Properties
- Configuring Notification for a Proxy

4.2.1 Creating System Properties

Oracle Identity Manager provides you with the capability of creating your own system properties. You can create system properties according to your requirements if you choose not to use any of the predefined system properties listed in "System Properties in Oracle Identity Manager" on page 4-1.

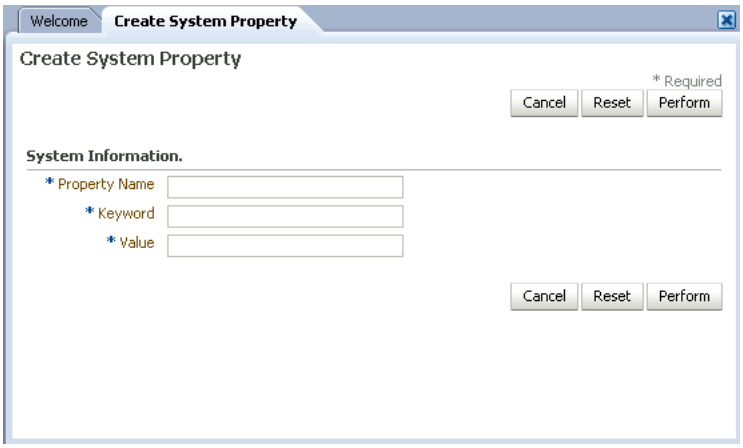
You can create a system property by using the Create System Property page in Oracle Identity Manager Administration. You can open this page only if you are authorized to create system properties.

While creating a system property, you specify values for the Property Name, Keyword, and Value fields. These values are saved in the PTY table of the Oracle Identity Manager database.

To create a system property:

1. Click the **System Management** tab, and then click **System Configuration.**
2. On the left pane, from the Actions menu, select **Create.** Alternatively, you can click the create icon on the toolbar. The Create System Property page is displayed, as shown in Figure 4–1:

Figure 4–1 Create System Property Page



3. On the Create System Property form, enter details of the system property. Table 4–3 describes the fields of this form.

Table 4–3 Fields of the Create System Property Form

Field	Description
Property Name	Enter a name of the system property.
Keyword	Enter a unique ID for the system property.
Value	Enter a value for the system property, for example, 4.

Note: Any special character (.) is not allowed in the beginning or end of Keyword fields while creating or updating a system property. In case of Value fields, special characters are allowed in the beginning or in the end.

4. Click **Perform** to create the system property. A message confirming that the system property has been created is displayed. For the new system property that is created, by default, the data level is set to 2 and login_required is set to true.

   After the system property is created, you can use SQL query to set values for the following system property fields that are automatically added to the system property recorded in the PTY table of the database:

   - **Data Level:** Every system property has a data level associated with it. The data level field determines the kind of operations that can be performed on a system property. Data levels are a means of specifying the operations that can be performed on a system property. For example, a data level value of 1 for a system property indicates that the system property can neither be modified nor deleted. The default value of this field is 2.

     The data level field cannot be modified by using the UI. It can only be modified by using a SQL script. Table 4–3 lists and describes the various data levels associated with a system property.

Table 4–4 Data Levels Associated with a System Property

Data Level	Description
0	Indicates that the system property can be modified or deleted
1	Indicates that the system property cannot be modified or deleted
2	Indicates that the system property can only be modified
3	Indicates that a system property can only be deleted

   - **Log In Required:** This field specifies whether or not a login is required to access the system property. The default value of this field is 1, which means that a login is required to access the system property. You can change the value of this field to 0 by using a SQL script.
   - **LKU_KEY:** This field determines the set of values that can be specified in the Value field of a system property. The default value of this field for a newly created system property is null. LKU_KEY is a column in the LKU table of the Oracle Identity Manager database. For a system property with non-null value in the LKU_KEY column, you can insert the values in this column from a predefined set of values that are in the LKV table. This is done by using a SQL script to include any valid LKU_KEY column value from the LKU table to associate multiple values with the system property. See step 7 for more details.

5. If you want to modify the data level of the system property, then run the following command:

   UPDATE PTY SET PTY_DATA_LEVEL=DATA_LEVEL_VALUE WHERE PTY_KEYWORD = SYSTEM_PROPERTY_KEYWORD;

   In this command:

   - *DATA_LEVEL_VALUE* is any value listed in the Data level column of Table 4–4.
   - *SYSTEM_PROPERTY_KEYWORD* is the unique ID for the system property that you entered in the Keyword field in Step 3.

   Note: Any special character (.) is not allowed in the beginning or end of Keyword fields while creating or updating a system property. In case of Value fields, special characters are allowed in the beginning or in the end.

6. If you want to modify the value of the Log In Required field, then run the following command:

   UPDATE PTY SET PTY_LOGINREQUIRED=LOGIN_REQUIRED_VALUE WHERE PTY_KEYWORD = SYSTEM_PROPERTY_KEYWORD;

   In this command:

   - *LOGIN_REQUIRED_VALUE* can take a value of either 0 or 1.

     If a login is required for accessing the system property, then enter `1`. Otherwise, enter `0`.

   - *SYSTEM_PROPERTY_KEYWORD* is the unique ID for the system property that you entered in the Keyword field in Step 3.

7. If you want to define the set of values that can be specified in the Value field of a system property, then run the following commands:

   a. Run the following command to insert a row into the LKU table:

   ```
   INSERT INTO LKU (LKU_KEY, LKU_LOOKUP_KEY, LKU_TYPE, LKU_GROUP,
   LKU_REQUIRED, LKU_TYPE_STRING_KEY, LKU_FIELD, LKU_DATA_LEVEL, LKU_CREATE,
   LKU_CREATEBY, LKU_UPDATE, LKU_UPDATEBY, LKU_NOTE, LKU_ROWVER) VALUEs
   (LKU_KEY_VALUE, LKU_LOOKUP_KEY_VALUE,...);
   ```

   For example, if you want to update a set of values for the Title field, then run the following INSERT statement:

   ```
   INSERT INTO LKU (LKU_KEY, LKU_LOOKUP_KEY, LKU_TYPE, LKU_GROUP,
   LKU_REQUIRED, LKU_TYPE_STRING_KEY, LKU_FIELD, LKU_DATA_LEVEL, LKU_CREATE,
   LKU_CREATEBY, LKU_UPDATE, LKU_UPDATEBY, LKU_NOTE, LKU_ROWVER) VALUES (201,
   Title, ...);
   ```

   Here, *LKU_KEY_VALUE* is 201 that uniquely identifies the record in the LKU table, and *LKU_LOOKUP_KEY_VALUE* is Title.

   Note: You must insert a record in the LKU table before inserting any record in the LKV table because the value of LKU_KEY is used in the LKV insert statement.

b. Run the following command to insert a row into the LKV table:

```
INSERT INTO LKV (LKV_KEY, LKU_KEY, LKV_ENCODED, LKV_DECODED, LKV_LANGUAGE,
LKV_COUNTRY, LKV_VARIANT, LKV_DISABLED, LKV_DATA_LEVEL, LKV_CREATE,
LKV_CREATEBY, LKV_UPDATE, LKV_UPDATEBY, LKV_NOTE, LKV_ROWVER) VALUES
(LKV_KEY_VALUE, LKU_KEY_VALUE, LKV_ENCODED_VALUE, LKV_DECODED_VALUE, ...);
```

For example, to define the set of values for the Title field as Mr, Ms, and Dr, run the following INSERT statements:

```
INSERT INTO LKV (LKV_KEY, LKU_KEY, LKV_ENCODED, LKV_DECODED, LKV_LANGUAGE,
LKV_COUNTRY, LKV_VARIANT, LKV_DISABLED, LKV_DATA_LEVEL, LKV_CREATE,
LKV_CREATEBY, LKV_UPDATE, LKV_UPDATEBY, LKV_NOTE, LKV_ROWVER) VALUES (1001,
201, 'Ms', 'Miss', ...);

INSERT INTO LKV (LKV_KEY, LKU_KEY, LKV_ENCODED, LKV_DECODED, LKV_LANGUAGE,
LKV_COUNTRY, LKV_VARIANT, LKV_DISABLED, LKV_DATA_LEVEL, LKV_CREATE,
LKV_CREATEBY, LKV_UPDATE, LKV_UPDATEBY, LKV_NOTE, LKV_ROWVER) VALUES (1002,
201, 'Mr', 'Mister', ...);

INSERT INTO LKV (LKV_KEY, LKU_KEY, LKV_ENCODED, LKV_DECODED, LKV_LANGUAGE,
LKV_COUNTRY, LKV_VARIANT, LKV_DISABLED, LKV_DATA_LEVEL, LKV_CREATE,
LKV_CREATEBY, LKV_UPDATE, LKV_UPDATEBY, LKV_NOTE, LKV_ROWVER) VALUES (1003,
201, 'Dr', 'Doctor', ...);
```

In this example:

- *LKV_KEY_VALUE* is 1001, 1002, and 1003 respectively that uniquely identifies the records in the LKV table
- *LKV_ENCODED_VALUE* is Ms, Mr, and Dr respectively
- *LKV_DECODED_VALUE* is Miss, Mister, and Doctor respectively

c. Run the following command to update the value of the LKU_KEY column in the PTY table:

UPDATE PTY SET LKU_KEY=LKU_KEY_COLUMN_IN_THE_LKV_TABLE WHERE PTY_KEYWORD = SYSTEM_PROPERTY_KEYWORD;

In this command:

- *LKU_KEY_COLUMN_IN_THE_LKV_TABLE* is the value of the LKU_KEY column in the LKV table.
- *SYSTEM_PROPERTY_KEYWORD* is the unique ID for the system property that you entered in the Keyword field in Step 3.

Note: If you want to view the changes in Oracle Identity Manager Advanced Administration, then you must run purge cache immediately after modifying a system property by using Microsoft SQL.

4.2.2 Purging Cache

Whenever you make any change to a system property by using any method other than from the Advanced Administration, you must run purge cache to get the changes reflected in Oracle Identity Manager:

Depending upon the operating system being used, run one of the following commands to clear the server cache:

- For Microsoft Windows:

  OIM_HOME\server\bin\PurgeCache.bat

- For UNIX:

  OIM_HOME/server/bin/PurgeCache.sh

4.2.3 Searching for System Properties

Oracle Identity Manager Advanced Administration allows you to perform the following types of search operations for system properties:

- Performing a Simple Search
- Performing an Advanced Search

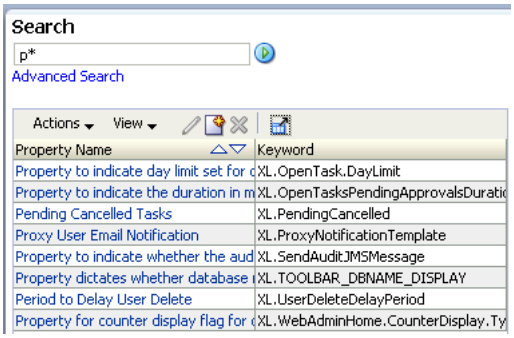
4.2.3.1 Performing a Simple Search

To perform a simple search for system properties:

1. In the Welcome page of Oracle Identity Manager Administration, under System Management, click **System Configuration**. Alternatively, you can click the **System Management** tab, and then click **System Configuration**.
2. In the left pane, enter a search criterion in the Search field for the system property that you want to search. You can include wildcard characters (*) in your search criterion.

   If you enter * in the Search field, then all the system properties are displayed. You can filter your search by combining characters with the wildcard characters. For example, to search all system properties starting with p, you can enter p* in the Search field.
3. Click the icon next to the Search field. A list of all system properties that meet the search criterion is displayed, as shown in Figure 4–2.

Figure 4–2 List of System Properties



   The search results table displays the system property names and keywords. You can click a property name to open the details for the system property.

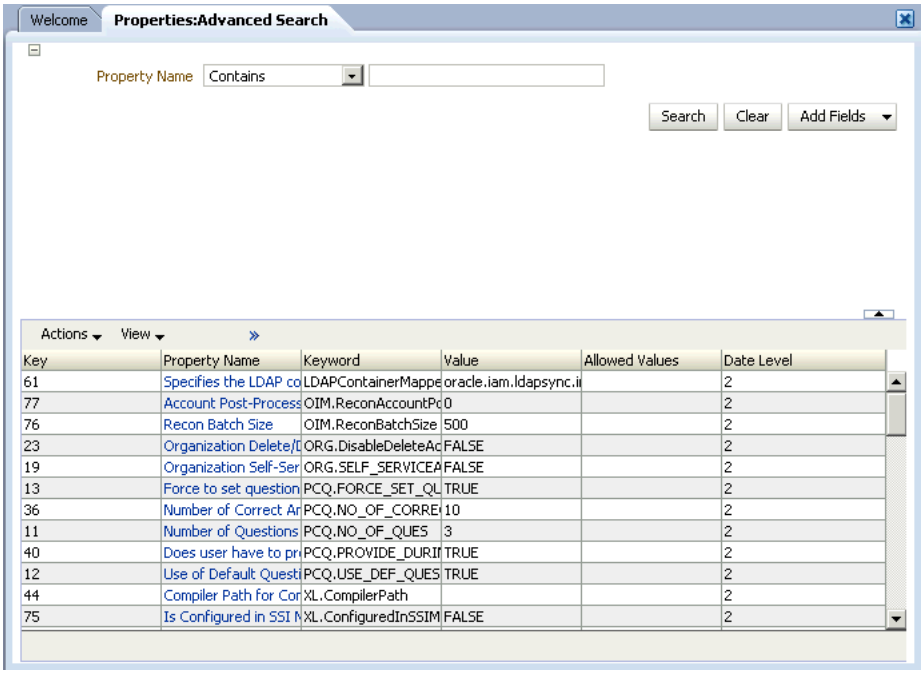
4.2.3.2 Performing an Advanced Search

To perform an advanced search for system properties:

1. In the left pane of the System Configuration section, click **Advanced Search**. The Properties: Advanced Search page is displayed.
2. In the list adjacent to the Property Name field, select a search condition.

3. In the Property Name field, enter a search criterion for the system property that you want to search. You can include wildcard characters (*) in your search criterion. Select the search conditions in the list adjacent to the fields. The search conditions include Not Contains, Not Begins With, Not Equals, Equals, Ends With, Not Ends With, Contains, and Begins With.
4. Click **Search**. The system properties that match the search criterion are displayed in the search results table, as shown in Figure 4–3:

Figure 4–3 Advanced Search Result



The search result displays key, property name, keyword, value, allowed value, and date level for each system property.

4.2.4 Modifying System Properties

A modify operation lets you modify an existing system property by using the System Property Detail page. If any system property is tagged with a set of allowed values, then you must specify a value from that set only.

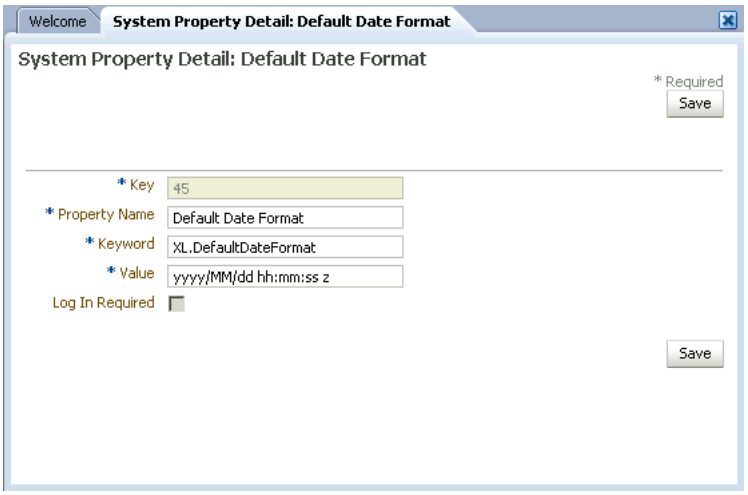
> **Note:** While modifying a system property that has multiple values attached to it, a message is displayed if the modified value is not part of the values defined in the LKU and LKV tables. For information about associating multiple values to a system property, see step 7 of "Creating System Properties" on page 4-14.

To modify a system property:

1. Search for the system property that you want to modify.
2. In the Property Name column of the search results table, click the system property that you want to modify.

   The System Property Details page is displayed, as shown in Table 4–4.

Figure 4–4 System Property Detail Page



3. If you want to modify the Property Name, keyword, and the Value fields, then perform Step 3 of "Creating System Properties" on page 4-14.
4. If you want to modify the Log In Required field, then perform Step 6 of "Creating System Properties" on page 4-14.
5. If you want to modify the Allowed Values column, then perform Step 7 of "Creating System Properties" on page 4-14.
6. If you want to modify the data level associated with a system property, then perform Step 5 of "Creating System Properties" on page 4-14.
7. Click **Save** to save the changes made.

   A message confirming that the system property has been modified is displayed.

4.2.5 Deleting System Properties

To delete a system property:

> **Note:** You can delete a system property only if the data level of that system property is set to either 0 or 3. While deleting a system property, a message is displayed if the data level associated with the system property is not appropriate. For a description of the data levels, see Table 4–4, " Data Levels Associated with a System Property".

1. Click the **System Management** tab and then click **System Configuration.**
2. On the left pane, search for the system property that you want to delete.
3. In the Property Name column of the search results table, select the system property that you want to delete.
4. From the Actions menu, select **Delete**. A message is displayed asking for confirmation. Click **OK**.
5. A message is displayed confirming that the system property has been deleted. Click **OK**.

4.2.6 Configuring Notification for a Proxy

Use the following steps to configure notification for a proxy:

1. Configure a new Email IT resource.
2. Create a new end user. (For example, create a "test1" user.)
3. Create a second end user. (For example, create a "test2" user.)
4. Assign the test1 user as a manager for the test2 user.
5. Specify your email ID for the test2 user, which enables you to receive notifications in your inbox.
6. Log in as test1 and navigate to the Oracle Identity Manager Self Service.
7. Select Profile, Proxies, and when the Proxies screen is displayed, add test2 as a proxy for the test1 user.

Note: If you successfully added the proxy, you (the test2 user in this case) will receive an email notification message similar to the following:

"You have been made the proxy for test1 test1[TEST1] from April 7, 2010 12:00:00 AM to April 30, 2010 12:00:00 AM".

5

Importing and Exporting Data Using the Deployment Manager

The Deployment Manager is a tool for exporting and importing Oracle Identity Manager configurations. The Deployment Manager lets you export the objects that constitute the Oracle Identity Manager configuration. Usually, you use the Deployment Manager to migrate a configuration from one deployment to another, for example, from a test to a production deployment, or to create a backup of your system.

> **Important:** To use Deployment Manager, JRE 1.4.2 must be installed on any computer that is running the Oracle Identity Manager Administrative and User Console.

You can save some or all of the objects in your configuration. This lets you develop and test your configurations in a test environment, and then import the tested objects into your production environment. You can export and import an object and all of its dependent and related objects at the same time. Alternatively, you can export and import each object individually.

The Deployment Manager allows you to retrieve configuration information from the source system, store the information in an XML file, and then import the information from the XML file to the target system. In Oracle Identity Manager 11*g* Release 1 (11.1.1), the Deployment Manager allows you to import data from the Oracle Identity Manager database, Meta Data Store (MDS) repository, or API repository. As a result, you can import all types of objects from these repositories, such as system properties, jobs, and scheduled tasks, which are not in the same repository. For example, you can import the scheduled tasks that are in the MDS repository instead of the database.

An object exported from one type of repository is imported to the same type of repository. For example, if a scheduled task is exported from the MDS repository, then the scheduled task is imported to the same repository, which is MDS, in the target system.

This chapter includes the following topics:

5.1 Features of the Deployment Manager

The Deployment Manager helps you to migrate Oracle Identity Manager deployments from one server environment to another, such as from a testing environment to a staging environment, or from a staging environment to a production environment.

The Deployment Manager enables you to:

- Update individual components of a deployment in different test environments
- Identify objects associated with components to be exported, so that those resources can be included
- Provide information about exported files
- Add comments

The Deployment Manager handles the following types of information:

- Roles
- Organizations
- Access policies
- Attestation processes
- Authorization policies
- User metadata
- Roles and organization metadata
- Scheduled tasks
- Scheduled jobs
- IT resources
- Resource objects
- Lookup definitions
- Process forms
- Provisioning workflows and process task adapters
- Data object definitions
- Rules
- Notification templates
- Generic Technology Connector (GTC) providers
- Error codes
- System properties
- E-mail definitions
- Event Handlers
- Password policies
- Generic Technology Connectors
- IT resource definition
- Request templates
- Request datasets

- Approval policies

The following are limitations of the Deployment Manager:

- **Merge Utility**: The Deployment Manager is not a merge utility.

  It cannot handle modifications done in both production and test environments. It replaces the object in the target system with that in the XML file.

- **Version Control Utility**: The Deployment Manager does not track versions of imported files, and does not provide rollback functionality.

  You can only use it as a means to move data between environments.

- **Code Moving**: The Deployment manager does not move JAR files in the JavaTasks directory or other locations. You must do this manually.

- **Custom Labels Move**: The Deployment Manager does not move labels defined in the customResources.properties file or the property files in the connectorResources directory. You must do this manually.

5.2 Exporting Deployments

You can export objects from your Oracle Identity Manager system and save them in an XML file. The Deployment Manager has an Export Wizard that lets you create your export file. Add objects by type, one type at a time, for example, roles, then forms, then processes, and so on.

If you select an object that has child objects or dependencies, you have the option to add them or not. After adding objects of one type, you can go back and add other objects to your XML files. When you have all the objects you want, the Deployment Manager saves them all at once in a single XML file.

Note:

- If a user belongs to a group to which the Export menu item has been assigned, then that user can export all the objects that are available for export, regardless of the permissions assigned to the user.

  A system administrator can export any object.

- When user-defined fields are associated with a specific resource object, during the export process one of the following events can occur:

  - If the user-defined fields contain values (entered information), then the Deployment Manager will consider them to be dependencies.

  - If the user-defined fields contain no values (the fields are blank), then the Deployment Manager will not consider them to be dependencies.

To export a deployment:

1. Login to Oracle Identity Manager Administration.
2. In the Welcome page, under System Management, under Deployment Manager, click **Export**. Alternatively, you can click the **System Management** tab, click **Deployment Manager**, and then click **Export**.

The Deployment Manager opens and the Search Objects page of the Export Wizard is displayed.

Note: To open the Deployment Manager by using Mozilla Firefox Web browser, an additional authentication dialog box might be displayed. Providing authentication in this dialog box allows access to the Deployment Manager. To avoid this additional authentication:

1. In Mozilla Firefox Web browser, from the Tools menu, select **Options**. The Options dialog box is displayed.
2. Click **Privacy**.
3. Select the **Accept third-party cookies** option.
4. Click **OK**.

The additional authentication is not required when the Deployment Manager is opened by using Microsoft Internet Explorer Web browser.

---

3. On the Search Objects page, select an object type from the menu, and enter search criteria. If you leave the criteria field blank, an asterisk (*) is displayed automatically to find all the objects of the selected type.

   All the objects supported by Deployment Manager for migration are available for exporting. See "Features of the Deployment Manager" on page 5-2 for the list of objects supported by Deployment Manager for migration.

4. Click **Search** to find objects of the selected type.

   To select an object, select the option of the object.

5. Click **Select Children**.

   The Select Children page is displayed with the selected objects and all of their child objects.

6. Select the child objects that you want to export.

   To select or remove an item, select the appropriate option.

   Click **Back** to go to the Search Objects page.

7. Click **Select Dependencies**.

   The Select Dependencies page is displayed with any objects required by the selected objects.

8. Select the dependent objects that you want to export.

   To select or remove an item, select the option of the item.

   Click **Back** to go to the Select Children page.

9. Click **Confirmation**.

   The Confirmation page is displayed.

10. Ensure that all the required items are selected, then click **Add for Export**.

    After you click **Add for Export**, you can still add more items to this export file.

    Click **Back** to go to the Search Objects page.

    The Add More page is displayed.

11. Use the wizard to add more items, or finish and exit the wizard. Select the appropriate option and click **OK**.

    If you select **Add more**, repeat Steps 2 through 7. Otherwise, the Export page is displayed.

    The Export page displays your current selections for export. Your selections have icons next to them that indicate what types of objects are selected. The Summary information pane shows the objects you are exporting. The Unselected Dependencies pane displays the list of dependent or child objects that you did not select for export.

12. Make any adjustments to your export file as follows:

    - Click **Reset** to clear the form.
    - Click **Legend** to see icon definitions.
    - Click **Add Object** to restart the wizard and add more items to your export file.

    To remove an object from the Current Selections list:

    - Right-click the object to remove and select **Remove** from the shortcut menu. If the object has child objects, then select **Remove including children** from the shortcut menu to remove the child objects all at the same time.
    - Click **Remove** to confirm. If the object is a child or dependency of a selected item, then it is added to the Unselected Children or Unselected Dependencies list.

    To add an object back to the Current Selections list from the Unselected Children or Unselected Dependencies list,

    a. Right-click the object, and select **Add**.

    b. Click **Confirmation**.

       The Confirmation page is displayed.

    c. Click **Add for Export**.

13. Click **Export**.

    The Add Description dialog box is displayed.

14. Enter a description for the file.

    This description is displayed when the file is imported.

15. Click **Export**.

    The Save As dialog box is displayed.

16. Enter a file name.

    You can browse to find a location.

17. Click **Save**.

    The Export Success dialog box is displayed.

18. Click **Close**.

5.3 Importing Deployments

Objects that were exported into an XML file by using the Deployment Manager can be imported into Oracle Identity Manager by using the Deployment Manager. You can

import all or part of the XML file, and you can import multiple XML files at once. The Deployment Manager ensures that the dependencies for any objects you are importing are available, either in the import or in your system. During an import, you can substitute an object you are importing for one in your system. For example, you can substitute a group specified in the XML file for a group in your system.

Note:

- If a user belongs to a group to which the Import menu item has been assigned, then that user must also have the necessary permissions for the objects that the user wants to import. Without these object-specific permissions, the Import operation fails. A system administrator can import any object.
- When more than 1000 resources, process definitions, parent forms, child forms, access policies, roles, and rules are imported by using the Deployment Manager, the size of the EIF table increases. The data can be truncated from this table by running a simple SQL query such as Delete from EIF.

This section discusses the following topics:

- Deployment Manager Actions on Reimported Scheduled Tasks
- Importing an XML File

Note: Before importing data that contains references to menu items, you must first create the menu items in the target system.

5.3.1 Deployment Manager Actions on Reimported Scheduled Tasks

A scheduled task is one of the objects that you can import by using the Deployment Manager. Typically, you import a scheduled task into your Oracle Identity Manager environment and later change the values of the scheduled attributes to meet your production requirements. However, if you import the same scheduled task a second time into the same Oracle Identity Manager server, the Deployment Manager does not overwrite the attribute values in the database. Instead, the Deployment Manager compares the attribute value of the reimported XML file to any corresponding attribute values in the database.

The following table summarizes the actions performed by the Deployment Manager during a scheduled task reimport:

Does the Scheduled Task have attribute values in the XML file being imported?	Are there any corresponding attribute values in the database?	Deployment Manager Action
Yes	No	Store attribute values in the database
No	Yes	Delete existing attribute values in the database
Yes	Yes (Newer attribute values indicated by time stamp)	No change in the database
Yes (New attribute values indicated by time stamp)	Yes	Update the database with the new attribute values

5.3.2 Importing an XML File

To import an XML file:

1. Login to Oracle Identity Manager Administration.
2. In the Welcome page, under System Management, under Deployment Manager, click **Import**. Alternatively, you can click the **System Management** tab, click **Deployment Manager**, and then click **Import**.

   The Deployment Manager opens and the Search Objects page of the Export Wizard is displayed.

   Note: To open the Deployment Manager by using Mozilla Firefox Web browser, an additional authentication dialog box might be displayed. Providing authentication in this dialog box allows access to the Deployment Manager. To avoid this additional authentication:

   1. In Mozilla Firefox, from the Tools menu, select **Options**. The Options dialog box is displayed.
   2. Click **Privacy**.
   3. Select the **Accept third-party cookies** option.
   4. Click **OK**.

   The additional authentication is not required when the Deployment Manager is opened by using Microsoft Internet Explorer Web browser.

3. Select a file.

   The Import dialog box is displayed.

4. Click **Open**.

   The File Preview page is displayed.

5. Click **Add File**.

   The Substitutions page is displayed

6. To substitute a name, click the **New Name** field adjacent to the item you want to replace, and enter the name.

   You can substitute only items that exist in the target system.

7. Click **Next**. If you are exporting an IT resource instance, then the Provide IT Resource Instance Data page is displayed. Otherwise, you are redirected to the Confirmation page.
8. Modify the values in the current resource instance and click **Next**, or click **Skip** to skip the current resource instance, or click **New Instance** to create a new resource instance.

   The Confirmation page is displayed.

9. Confirm that the information displayed on the Confirmation page is correct.

   To go back and make changes, click **Back**, or click **View Selections**.

   The Deployment Manager Import page displays your current selections.

   The Import page also displays icons next to your current selections. The icons indicate what types of objects are selected. The icons on the right indicate the status of your selections. The file names of any selected files, summary

information about the objects you are importing, and substitution information are displayed on the left side of the page. On the right, the **Objects Removed from Import** list displays any objects in the XML file that will not be imported.

10. Make any of the following adjustments:
    - Click **Reset** to clear the form.
    - Click **Legend** to see icon definitions.
    - To remove an object from the Current Selections list, right-click the object, select **Remove** from the shortcut menu, and then click **Remove** to confirm that you want to remove the object.

      If the object has child objects, then select **Remove including children** from the shortcut menu to remove all the child objects at the same time. The item is added to the Objects Removed From Import list.
    - To add an item back to the Current Selections list, right-click the list, and click **Add**.

      If the object has child objects, then select **Add including children** from the shortcut menu to add all the child objects at the same time.
    - To make substitutions, click **Add Substitutions**.
    - To add objects from another XML file, click **Add File** and repeat Steps 2 through 7.
    - Click **Show Information** to see information about your imported information.

      The Information page is displayed.

      To see more information, select the **Show Info Level Messages** option, and then click **Show Messages**. Click **Close** to close the Information page.
11. To import the current selections, click **Import**.

    A confirmation dialog box is displayed.
12. Click **Import**.

    The Import Success dialog box is displayed.
13. Click **OK**.

    The objects are imported into Oracle Identity Manager.

5.4 Horizontal Migration of Entities

The Deployment Manager is used for performing migration of metadata entities from an Oracle Identity Manager deployment to another. However, for Oracle Identity Manager 11*g* Release 1 (11.1.1), there are other non-metadata entitles that are not supported by the Deployment Manager. These entities include custom resource bundles and plug-ins. Therefore, a complete migration of entities is performed by using a command-line utility, which is the horizontal migration utility, along with the Deployment Manager.

The horizontal migration command-line utility supports the migration of the following metadata entities that are not supported by the Deployment Manager:

- Custom resource bundle
- Plug-ins

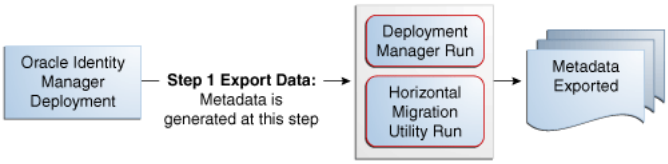
The migration of metadata entities take place in the following steps:

1. **Export data:** When data from an Oracle Identity Manager deployment is exported by running the Deployment Manager and the horizontal migration command-line utility, a set of artifacts are generated. The Deployment Manager generates XML files, and the horizontal migration utility generates binaries and XML files.

   Note: Deployment Manager supports the migration of all the entities in the form of XML. The command-line utility supports the migration of binaries, which are entities that are not exportable and importable in the form of XML.

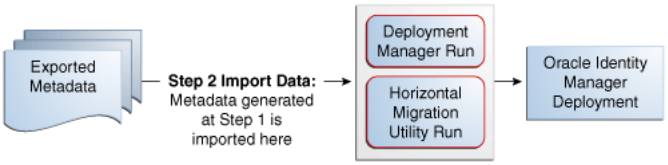
   Figure 5–1 shows the exporting of data:

Figure 5–1 Exporting Migration Data



2. **Import data:** The Deployment Manager and the horizontal migration utility are run to import the metadata on the second Oracle Identity Manager deployment, as shown in Figure 5–2:

Figure 5–2 Importing Migration Data



The horizontal migration utility is used to migrate the entities that are not supported by the Deployment Manager. This section describes the export and import of entities by using the horizontal migration utility in the following sections:

- Creating a Backup of the Existing Entities
- Running the Horizontal Migration Utility
- Data Migration for Supported Entities
- Horizontal Migration Report

5.4.1 Creating a Backup of the Existing Entities

Before performing the migration, create a backup of the existing entities in the Oracle Identity Manager deployment. If you are importing any entity, then create a backup of the existing ones so that you can roll back if required.

To create the backup, use the horizontal migration utility in the export mode to extract the existing entities. See "Running the Horizontal Migration Utility" on page 5-10 for information about running the utility in export mode.

5.4.2 Running the Horizontal Migration Utility

When you run the horizontal migration utility in EXPORT mode, a ZIP file is created that contains all the artifacts of the entities to be migrated. You must migrate the ZIP file into the second deployment where the data is to be imported back. When you run the utility in IMPORT mode, the contents of the ZIP file is extracted in a temporary location and all the artifacts are imported in the Oracle Identity Manager deployment. The configuration in the properties file controls the export and import. All the configurations in the file are defined at runtime.

In the EXPORT mode, you run the exportMetaData.sh or exportMetaData.bat script, which is in the *OIM_HOME*/bin directory.

To run the horizontal migration utility in EXPORT mode:

1. Check the location of the Config.xml file. The Config.xml file contains the filter criterion for filtering the entities for export. You can modify this file to provide custom filters.

   Save the Config.xml file before running the utility.

2. Run the exportMetaData.sh or exportMetaData.bat script after specifying the following input parameters in the utility script:

   - Username to connect to Oracle Identity Manager
   - Password to connect to Oracle Identity Manager
   - JNDI URL to connect to Oracle Identity Manager
   - Context to connect to Oracle Identity Manager
   - Destination path for the package to be exported
   - Configuration file that you must create with the definition of the parameters and filtering criteria for the Export of the metadata

     The following is a sample configuration XML file:

```
<?xml version='1.0' encoding='UTF-8'?>
<MigrationDetails Operation ="Export">
        <entityDetails>
              <EntityType>Jars</EntityType>
              <FilteringCriteria>
                     <Attribute>
                          <Name>JarName</Name>
                          < Filter >*</ Filter >
                     </Attribute>
              </FilteringCriteria>
        </entityDetails>

        <entityDetails>
              <EntityType>Plugins</EntityType>
              <FilteringCriteria>
                     <Attribute>
                          <Name>PluginName</Name>
                          < Filter >*</ Filter >
                     </Attribute>
              </FilteringCriteria>
        </entityDetails>

        <entityDetails>
              <EntityType>CustomResourceBundles</EntityType>
              <FilteringCriteria>
```

```
                        <Attribute>
                             <Name>FileName</Name>
                             <Filter>*</Filter>
                        </Attribute>
                   </FilteringCriteria>
              </entityDetails>
         </MigrationDetails>
```

The configuration file supports three entity types: Jars, Plug-ins, and CustomResourceBundles. For each entity type, the following filters are supported:

- **Jars:** Jar_Type , Jar_Name
- **Plugins:** Plugin_Name
- **CustomResourceBundles:** Resource_Type , Resource_Name

- Temporary location to keep the files temporarily before packaging for export
- LogFileLocation path where log file is to be generated

3. Specify the following when prompted:
   - Oracle Identity Manager administrator user name
   - Oracle Identity Manager administrator password
   - Server URL: t3://localhost:*PORT_NUMBER*
4. Verify the export list that is displayed.
5. When prompted for confirmation, enter YES.
6. Verify the export. All the listed items are exported to the destination provided as input. Check the contents of the ZIP package that is created at the destination.

In the IMPORT mode, you run the importMetaData.sh or importMetaData.bat script, which is in the *OIM_HOME*/bin directory.

To run the horizontal migration utility in IMPORT mode:

1. Before running the utility, run the client targets by using the following commands:

```
ant fullbuild XellerateClient.view-install
ant assemble-ear client-archive
```

2. Run the importMetaData.sh or importMetaData.bat script after specifying the following input parameters in the utility script:
   - Username to connect to Oracle Identity Manager.
   - Password to connect to Oracle Identity Manager.
   - JNDI URL to connect to Oracle Identity Manager.
   - Context to connect to Oracle Identity Manager.
   - Path of the package to be imported.
   - Configuration file updated with the information about items to be imported. If this configuration is not used in import, then the entire content of the package is imported.
   - Temporary location where the package is to be extracted before importing.
3. Specify the following when prompted:
   - Oracle Identity Manager administrator username

- Oracle Identity Manager administrator password
- Server URL: t3://localhost:*PORT_NUMBER*

4. Verify the import list that is displayed.
5. When prompted for confirmation, enter `YES`.
6. Verify the import. All the items in the package are imported to the application. Check if the import utility creates the entries corresponding to all the package contents in the database tables if you have access to the schema. Otherwise, check the utility output log in the application to verify if all contents have been successfully imported.

5.4.3 Data Migration for Supported Entities

This section describes the migration of the following entities:

- Custom Resource Bundle
- Plug-ins

5.4.3.1 Custom Resource Bundle

Oracle Identity Manager stores localized versions of text strings that appear in the user interface in resource bundles. In addition to the default resource bundles, the custom resource bundles, which are stored in Oracle Identity Manager database, can be imported and exported by using the horizontal migration utility.

The custom resource bundles are available in the following property files:

- oim.ear/xlWebApp.war/WEB-INF/classes/xlRichClient_*.properties
- Agent_*.properties for each feature in the deployment

5.4.3.2 Plug-ins

Plug-ins are stored in Oracle Identity Manager database. The horizontal migration utility migrates the binaries from plug-in database store of one deployment to another.

> **See Also:** "Working with the Plug-in Framework" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for information about defining and using plug-ins

5.4.4 Horizontal Migration Report

After the horizontal migration utility is run, a report is generated that contains the following information:

- All the entities migrated by using this utility
- Status of overall export and import of metadata
- Errors that occurred during the import of metadata

The following is a sample report:

```
Plugins :
Failed to process element Plugin1".
Exception details are java.io.FileNotFoundException: C:\Plugin1.zip (The system
cannot find the path specified) at java.io.FileInputStream.open(Native Method)at
java.io.FileInputStream.<init>(Unknown Source)at java.io.FileReader.<init>(Unknown
Source)at file.main(file.java:13)
```

5.5 Best Practices Related to Using the Deployment Manager

The following are some of the suggested practices and pitfalls to avoid while by using Deployment Manager:

5.5.1 Export System Objects Only When Necessary

You should export or import system objects, for example, Request, Xellerate User, and System Administrator, only when it is absolutely necessary. Exporting system objects from the testing and staging environments into production can cause problems. If possible, exclude system objects when exporting or importing data.

You may want to export or import system objects when, for example, you define trusted source reconciliation on Xellerate User resource objects.

Caution: The Deployment Manager keeps track of imported components and structures, but not of completed imports. After an import is completed, you cannot roll it back to a previous version. A new import is required.

5.5.2 Export Related Groups of Objects

Oracle recommends that you use the Deployment Manager to export sets of related objects. A unit of export should be a collection of logical items that you want to group together.

Avoid exporting everything in the database in one operation, or exporting items one at a time. For example, suppose that you manage an integration between Oracle Identity Manager and a target system that includes processes, resource objects, adapters, IT resource type definitions, IT resource definitions, scheduled tasks, and so on. For this environment, you should create groups of related objects before exporting.

For example, if you use the same e-mail definitions in multiple integrations, you should export the e-mail definitions as one unit, and the integrations as a different unit. This enables you to import changes to e-mail definitions independently of target system integration changes. Or, if multiple resources use the same IT resource type definition, you can export and import the type definition separately from other data.

You can import one or more sets of exported data at a time. For example, you can import a resource object definition, an e-mail definition, and an IT resource type definition in a single operation.

5.5.3 Group Definition Data and Operational Data Separately

You must group and export definition data and operational data separately.

You configure definition data in the testing and staging environment. Definition data includes resource objects, processes, and rules.

You typically configure operational data in the production environment. Operational data includes groups and group permissions. The testing and staging servers usually do not include this data.

By grouping data according to where it is changed, you know what data goes to testing and staging, and what goes to production. For example, if approval processes are changed in production, you should group approval processes and export them with other operational data.

5.5.4 Use Logical Naming Conventions for Versions of a Form

You often revise forms multiple times before exporting them. Avoid generic names, for example, "v23," to differentiate among versions of a form. Create meaningful names, for example, "Before Production" or "After Production Verification." Do not use special characters, including double quotation marks, in version names.

5.5.5 Export Root to Preserve a Complete Organizational Hierarchy

When you export a leaf or an organization in an organizational hierarchy, only one dependency level is exported. To export a complete organizational hierarchy, you must export the root of the hierarchy.

5.5.6 Provide Clear Export Descriptions

The Deployment Manager records some information automatically, for example, the date of the export, who performed the export, and the source database. You must also provide a meaningful description of the content of the export, for example, "resource definition after xxx attributes added in reconciliation." This informs the importer of the file of the contents of the data being imported.

5.5.7 Check All Warnings Before Importing

When importing information to the production environment, check all the warnings before completing the import operation. Treat each warning seriously.

5.5.8 Check Dependencies Before Exporting Data

The wizard in the top right pane shows resources that must be available in the target system.

Consider the following types of dependencies:

- If the resources are already available in the target system, they do not need to be exported.
- If the resources are new (not in the target system), they must be exported.
- If the target system does not include the resources, such as lookups, IT resource definitions, or others that are reused, then record the data and export it in a separate file so it can be imported if necessary.

Note: When you export a resource, groups with Data Object permissions on that form are not exported with the resource.

5.5.9 Match Scheduled Task Parameters

Scheduled tasks depend on certain parameters to run properly. You can import scheduled task parameters to the production server. Table 5–1 shows the rules for determining how to import scheduled tasks. Note that parameters may be available for tasks that no longer reside on the target system.

Table 5–1 Parameter Import Rules

Parameter Exists in Target System	Parameter Exists in the XML File	Action Taken
Yes	No	Remove the parameter from the target system.
No	Yes	Add the parameter and current value from the XML file.
Yes	Yes	Use the more recent value of the parameter.

5.5.10 Compile Adapters and Enable Scheduled Tasks

After an import operation, the adapters are set to recompile and the scheduled tasks are disabled. After importing the classes and adjusting the task attributes, manually recompile the adapters and enable the scheduled tasks.

5.5.11 Export Entity Adapters Separately

Entity adapters are modified to bring just the entity adapter, not its usage. If you want to export the usage of an entity adapter, you must separately export each use with a data object by exporting the data object. If you export a data object, all the adapters and event handlers attached to the object along with the permissions on the object are exported. You must pay special attention when exporting data objects. For example, to export a form, you should also add the data object corresponding to the form. This ensures that the associated entity adapters can use the form.

5.5.12 Check Permissions for Roles

When you export roles, the role permissions on different data objects are also exported. However, when you import data, any permissions for missing data objects are ignored. If the role is exported as a way of exporting role permission setup, then check the warnings carefully to ensure that permission requirements are met. For example, if a role has permissions for objects A, B, and C, but the target system only has objects A and B, the permissions for object C are ignored. If object C is added later, the role permissions for C must be added manually, or the role must be imported again.

When you export role that have permissions for viewing certain reports, ensure that the reports exist in the target environment. If the reports are missing, then consider removing the permissions before exporting the role.

5.5.13 Back Up the Database

Before you import data into a production environment, back up the database. This enables you to restore the data if anything goes wrong with the import. Backing up the database is always a good precaution before making significant changes.

Note: When you import forms and user-defined fields, you add entries to the database. These database entries cannot be rolled back or deleted. Before each import operation, ensure that the correct form version is active.

5.5.14 Import Data When the System Is Quiet

You cannot complete an import operation in a single transaction because it includes schema changes. These changes affect currently running transactions on the system. To limit the effect of an import operation, temporarily disable the Web application for general use and perform the operation when the system has the least activity, for example, overnight.

5.5.15 Update the SDK Table

The SDK table contains metadata definitions for user-defined data objects. When you import data from an XML file into the SDK table, the values in the SDK_SCHEMA column might be modified with the schema name of the source system where the XML file was created. For this reason, after you import data from an XML file into the SDK table, you must check the schema name in the SDK_SCHEMA column, and if necessary, manually change it to the schema name on the target system where the Oracle Identity Manager database is running. To update the schema name in the SDK_SCHEMA column, run a SQL query similar to the following with SQL*Plus on Oracle Database installations or with SQL Query Analyzer on Microsoft SQL Server installations:

```
UPDATE SDK SET SDK_SCHEMA='target system schema name'
```

If you do not update the schema name in the SDK_SCHEMA column, an error similar to the following might be generated when you import other XML files that modify user-defined field (UDF) definitions:

```
CREATE SEQUENCE UGP_SEQ
java.sql.SQLException: ORA-00955: name is already used by an existing object
```

5.5.16 Remove Data Object Fields Before Importing Event Handlers as Dependencies

The Deployment Manager does not import event handlers that include data object fields if the event handlers are imported as dependencies. For this reason, you must remove the data object fields from any event handlers that you want to import as dependencies with the Deployment Manager.

5.6 Best Practices for Using the Horizontal Migration Utility

The following are some of the suggested practices and pitfalls to avoid while by using the horizontal migration utility:

- Export system objects only when necessary. See "Export System Objects Only When Necessary" on page 5-13.
- Export related groups of objects. See "Export Related Groups of Objects" on page 5-13.
- Check all listing before importing or exporting. See "Check All Warnings Before Importing" on page 5-14.
- Create a backup of the database. "Back Up the Database" on page 5-16.
- Provide filter criteria as specific as possible in the Config.xml file. See step 3 in "Running the Horizontal Migration Utility" on page 5-10.

  For example, consider the following filter criteria:

```
<entityDetails>
      <EntityType>CustomResourceBundles</EntityType>
      <FilteringCriteria>
            <Attribute>
                  <Name>FileName</Name>
                  <Filter>*</Filter>
            </Attribute>
      </FilteringCriteria>
</entityDetails>
```

  Instead of using the asterisk (*) wildcard character as the filter criteria, specify a file name or combine a file name with wildcard characters, such as `<Filter>*.properties</Filter>`.

6

Installing Connectors

You use a predefined connector to integrate Oracle Identity Manager with a specific third-party application. This chapter discusses the procedure to install predefined connectors.

> **Note:** The predefined connectors are distributed in the Oracle Identity Manager Connector Pack, independent from the Oracle Identity Manager core server release.
>
> See the Oracle Identity Manager Connector Pack documentation to determine whether or not you can install the required release of the connector by using the Install Connector feature of Oracle Identity Manager Administration.

This chapter is divided into the following sections:

6.1 Overview of the Connector Installation Process

The installation of most predefined connectors requires you to perform some or all of the following tasks:

1. Verify the installation requirements.
2. Configure the target system.
3. Copy the external code files to a directory on the Oracle Identity Manager server.
4. Configure the Oracle Identity Manager server.
5. Import the connector XML files.
6. Configure reconciliation.
7. Configure provisioning.
8. Configure Secure Sockets Layer (SSL).

Of these tasks, the Oracle Identity Manager Administration can be used to perform the following:

- Copying the connector files and external code files to directories on the Oracle Identity Manager server

- Importing the connector XML files
- Compiling adapters (which is part of the procedure to configure provisioning)

Note: You must manually perform the remaining tasks. For instructions on performing these tasks, see the connector-specific documentation in the Oracle Identity Manager Connector Pack documentation library.

6.2 Installing a Predefined Connector

To install a predefined connector:

1. Log in to the Administrative and User Console.
2. Go to Advanced Administration.
3. In the Welcome page, under Deployment Manager, click **Install Connector**.
4. From the Connector List list, select the connector that you want to install. This list displays the names and release numbers of connectors whose installation files you copy into the default connector installation directory:

   OIM_HOME/ConnectorDefaultDirectory

   If you have copied the installation files into a different directory, then:

   a. In the **Alternative Directory** field, enter the full path and name of that directory.
   b. To repopulate the list of connectors in the Connector List list, click **Refresh**.
   c. From the Connector List list, select the connector that you want to install.
5. Click **Load**.

   Information about the following is displayed:

   - Connector installation history

     The connector installation history is information about previously installed releases of the same connector.

   - Connector dependency details

     There are some connectors that require the installation of some other connectors before you can start using them. For example, before you use the Novell GroupWise connector, you must install the Novell eDirectory connector. Novell eDirectory is called the **dependency connector** for Novell GroupWise.

     The connector dependency details include the list of connectors that must be installed before you install the selected connector. These details also include information about any dependency connectors that are already installed, and whether or not any of the installed dependency connectors must be upgraded.

     You must ensure that the correct versions of dependency connectors are installed before you proceed with the connector installation.
6. To start the installation process, click **Continue**.

   The following tasks are performed in sequence:

   a. Configuration of connector libraries

b. Import of the connector XML files (by using the Deployment Manager)

c. Compilation of adapters

On successful completion of a task, a check mark is displayed for the task. If a task fails, then an X mark and a message stating the reason for failure are displayed. Depending on the reason for the failure, make the required correction and then perform one of the following steps:

- Retry the installation by clicking **Retry.**
- Cancel the installation and begin again from Step 1.

One of the reasons for installation failure could be a mismatch between information about files and directory paths in the configuration XML file and the actual files and directory paths. If this happens, then an error message is displayed.

For example, suppose the actual name of the JAR file for reconciliation is `recon.jar`. If the name is provided as `recon1.jar` in the configuration XML file, then an error message is displayed.

If such an error message is displayed, then perform *any one* of the following steps:

- Make the change in the configuration XML file, and then retry installation from the Step 1: Select Connector to Install page onward.

  In the example described earlier, change the name of the JAR file to `recon.jar` in the configuration XML file, and then retry installation from the Step 1: Select Connector to Install page onward.

- Make the change in the actual name or path of the file or directory, and then use the Retry option.

  In the example described earlier, change the name of the JAR file to `recon1.jar` and then click the **Retry** button.

7. Some connectors have user-defined fields (UDF) that need additional configuration so that they can be searched on. The Active Directory connector field USR_UDF_OBGUID is an example of this type of field. To make a UDF searchable

a. Go to the User Attributes page, select the UDF attribute, and set the Searchable property to true. See "Modifying Entity Attributes" on page 13-8 for information about editing the attributes specific to user entity.

b. Create an authorization policy to allow searching on the UDF. The policy should set Search User and View User Details permissions. See "Creating Custom Authorization Policies" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about creating authorization policies.

8. If all three tasks of the connector installation process are successful, then a message indicating successful installation is displayed. In addition, a list of the steps that you must perform after the installation is displayed. These steps are as follows:

a. Ensuring that the prerequisites for using the connector are addressed

---

Note: At this stage, run the PurgeCache utility to load the server cache with content from the connector resource bundle in order to view the list of prerequisites. See "Purging the Cache" on page 26-3 for information about running the `PurgeCache` utility.

There are no prerequisites for some predefined connectors.

---

b. Creating an IT resource for the connector

The IT resource type is displayed. You must create an IT resource of the specified type.

See Also: "Creating IT Resources" for information about how to create IT resources in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*

c. Configuring the scheduled tasks that are created when you installed the connector

The names of the scheduled tasks that are created during the XML file import process are displayed. You must configure these scheduled tasks.

See Also: Chapter 2, "Managing Scheduled Tasks" for information about configuring scheduled tasks

---

Note: You can also access links to the Oracle Identity Manager Administrative and User Console pages for creating the IT resource and configuring the scheduled tasks by expanding the **Resource Management** menu on the left navigation pane of the console.

6.3 Using Custom Connectors

You can create a custom connector to link Oracle Identity Manager and a target system that has no predefined connector by using Generic Technology Connector (GTC). For detailed information about GTC, see *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

Part II

System Management

This part describes the system management tasks in Oracle Identity Manager.

This part contains the following chapters:

7

Starting and Stopping Servers

Most Oracle Identity Manager feature configurations, such as password policy create and update, need the restart of the server for the changes to take effect. This chapter provide procedures to start/stop Oracle WebLogic Servers.

You can perform all start and stop operations for managed WebLogic Servers either from command prompt or from Oracle WebLogic Server Administration Console or Enterprise Manager control.

The following sections are given only for reference purpose. See Oracle *WebLogic Server Administrator Guide* for detailed information.

> **Note:** Node Manager must be running in order to use the Oracle WebLogic Server Administration Console or Enterprise Manager Control for controlling (start/stop) Oracle Identity Manager WebLogic managed servers and SOA WebLogic managed servers.

- Configuring the Node Manager
- Starting the Node Manager
- Starting or Stopping WebLogic Administration Server
- Starting or Stopping WebLogic Managed Servers

You can perform all start and stop operations either from command prompt or from Oracle WebLogic Server Administration Console.

> **Note:** Node Manager must be running before you can start and stop administration server, managed server, SOA server, and BPEL server through Oracle WebLogic Server Administration Console.

7.1 Configuring the Node Manager

After installing and configuring Oracle Identity Manager and SOA servers, you must configure node manager for using it with WebLogic Administration Console or Enterprise Manager control. This configuration is to be done only once.

To configure node manager, you must set `StartScriptEnabled=true` in the nodemanager.properties file. To do so, run following script:

For UNIX:

MIDDLEWARE_HOME/oracle_common/common/bin/setNMProps.sh

For Microsoft Windows:

MIDDLEWARE_HOME\oracle_common\common\bin\setNMProps.cmd

7.2 Starting the Node Manager

To start the Node Manager:

1. Navigate to *WL_HOME*/server/bin.
2. At the command prompt, enter:

   ```
   ./startNodeManager
   ```

7.3 Starting or Stopping WebLogic Administration Server

To start or stop the WebLogic Administration Server:

1. Navigate to *DOMAIN_HOME*/bin.

   Note:

   - For Linux Install you have only "./startWebLogic.sh" and you do not have "startWebLogic.cmd" in the bin folder.
   - For Microsoft Windows Install we have both "./startWebLogic.sh" and "startWebLogic.cmd" in the bin folder.

2. To start the server, enter the following:

   For UNIX:

   ```
   ./startWebLogic.sh
   ```

   For Microsoft Windows:

   ```
   startWebLogic.cmd
   ```

   To stop the server, enter the following:

   For UNIX:

   ```
   ./stopWebLogic.sh
   ```

   For Microsoft Windows:

   ```
   stopWebLogic.cmd
   ```

7.4 Starting or Stopping WebLogic Managed Servers

This section contains the following topics:

- Starting or Stopping the Managed Servers By Using Command Prompt
- Starting or Stopping the Managed Server Using Oracle Enterprise Manager Console
- Starting or Stopping Servers By Using Oracle WebLogic Server Administration Console

7.4.1 Starting or Stopping the Managed Servers By Using Command Prompt

To start or stop the managed servers using command prompt:

1. Navigate to the *DOMAIN_HOME*/bin/ directory.
2. To start the server, enter the following at the command prompt:

   For UNIX:

   `./startManagedWebLogic.sh` *`MANAGED_SERVER_NAME ADMIN_SERVER_URL`*

   For example:

   ```
   startManagedWebLogic.sh oim_server1
   http://mywlsadminhost.mycompany.com:7001
   ```

   ```
   startManagedWebLogic.sh soa_server1
   http://mywlsadminhost.mycompany.com:7001
   ```

   For Microsoft Windows:

   `startManagedWebLogic.cmd` *`MANAGED_SERVER_NAME ADMIN_SERVER_URL`*

   To stop the server, enter the following at the command prompt:

   For UNIX:

   `./stopManagedWebLogic.sh` *`MANAGED_SERVER_NAME ADMIN_SERVER_URL`*

   For Microsoft Windows:

   `stopManagedWebLogic.cmd` *`MANAGED_SERVER_NAME ADMIN_SERVER_URL`*

   For example:

   stopManagedWebLogic.cmd oim_server1
   http://mywlsadminhost.mycompany.com:7001

   stopManagedWebLogic.cmd soa_server1
   http://mywlsadminhost.mycompany.com:7001

7.4.2 Starting or Stopping the Managed Server Using Oracle Enterprise Manager Console

In order to use the Oracle Enterprise Manager Console to control managed servers, Node Manager must be running on the computer.

To start or stop the managed server using Oracle Enterprise Manager Console:

1. Log in to the Oracle Enterprise Manager Console.
2. Navigate to **Weblogic Domain**, **Domain Name**, ***SERVER_NAME***.
3. Right click, and navigate to **Control**.
4. Click **Start Up** to start the server.

   Click **Shutdown** to stop the server.

7.4.3 Starting or Stopping Servers By Using Oracle WebLogic Server Administration Console

To start or stop servers by using Oracle WebLogic Administration Console:

1. Log in to the Oracle WebLogic Server Administration Console.

2. On the left pane, under Domain Structure, select **Environment**, **Servers**.
3. On the right pane, under Summary of Servers, click the **Control** tab.
4. Select the server name.
5. Click **Start** to start the server.

   Click **Shutdown** to shutdown the server.

8

Enabling System Logging

Oracle Identity Manager uses two logging services, Oracle Diagnostic Logging (ODL) and Apache log4j.

Oracle Identity Manager logging is primarily done with ODL. Apache log4j is only used with third-party applications, such as Nexaweb for Deployment Manager and Workflow Designer, and OSCache for caching.

This chapter contains the following sections:

8.1 Logging in Oracle Identity Manager By Using ODL

Oracle Diagnostic Logging (ODL) is the principal logging service used by Oracle Identity Manager. For ODL logging to work, both loggers and log handlers need to be configured. Loggers send messages to handlers, and handlers accept messages and output them to log files.

Each Oracle Identity Manager module has its own logger that can be configured independently to send different amounts of information to one or more log handlers. Table 8–2, " Oracle Identity Manager Loggers" lists the more than twenty different Oracle Identity Manager loggers that can be configured to send messages to log handlers.

You can output more or less information to a log by adjusting the level attribute for each logger. To select a logging level, choose from one of five message types (INCIDENT_ERROR, ERROR, WARNING, NOTIFICATION, and TRACE). Each message type can also take a numeric value between 1 (highest severity) and 32 (lowest severity) that you can use to further restrict the volume of messages that a logger will output. Table 1 on page 2 lists the message type and level combinations that are used most often.

Log handlers specify the target where log messages should appear. For example, log handlers can write messages to the console, to various log files, and to additional outputs.

This section contains the following topics:

- Sample ODL Log Output

8.1.1 Message Types and Levels

ODL recognizes five message types: INCIDENT_ERROR, ERROR, WARNING, NOTIFICATION, and TRACE. Each message type can also take a numeric value between 1 (highest severity) and 32 (lowest severity) that you can use to further restrict message output.

When you specify a message type, ODL returns all messages of that type, as well as the messages that have a higher severity. For example, if you set the message type to WARNING, ODL also returns messages of type INCIDENT_ERROR and ERROR.

Message types and levels are described in greater detail in "Setting the Level of Information Written to Log Files" of the *Oracle Fusion Middleware Administrator's Guide*. Table 8–1 lists the diagnostic message types that you can use most often with Oracle Identity Manager.

Table 8–1 Oracle Identity Manager Diagnostic Message Types

Message Type and Numeric Value	Description
INCIDENT_ERROR:1	A serious problem that may be caused by a bug in the product and that should be reported to Oracle Support. Examples are errors from which you cannot recover.
ERROR:1	A serious problem that requires immediate attention from the administrator and is not caused by a bug in the product. An example is if Oracle Fusion Middleware cannot process a log file, then you can correct the problem by fixing the permissions on the document.
WARNING:1	A potential problem that should be reviewed by the administrator. Examples are invalid parameter values or a specified file does not exist.
NOTIFICATION:1	A major lifecycle event such as the activation or deactivation of a primary sub-component or feature. This is the default level for NOTIFICATION.
NOTIFICATION:16	A finer level of granularity for reporting normal events.
TRACE:1	Trace or debug information for events that are meaningful to administrators, such as public API entry or exit points.
TRACE:16	Detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.
TRACE:32	Very detailed trace or debug information that can help Oracle Support diagnose problems with a particular subsystem.

8.1.2 Log Handler and Logger Configuration

Both log handlers and loggers can be configured by editing logging.xml, which is located in:

DOMAIN_NAME/config/fmwconfig/servers/*SERVER_NAME*/logging.xml

Here, *DOMAIN_NAME* and *SERVER_NAME* are the domain name and server name respectively specified during the installation of Oracle Identity Manager.

The logging.xml file has a <log_handlers> configuration section, followed by a <loggers> configuration section. Each log handler is defined within the <log_handlers> section, and each logger is defined within the <loggers> section.

The file has the following basic structure:

```
<logging configuration>
  <log_handlers>
    <log_handler name='console-handler' level="NOTIFICATION:16"></log_handler>
    <log_handler name='odl-handler'></log_handler>
    <!--Additional log_handler elements defined here....-->
  </log_handlers>
  <loggers>
    <logger name="example.logger.one" level="NOTIFICATION:16">
      <handler name="console-handler"/>
    </logger>
    <logger name="example.logger.two" />
    <logger name="example.logger.three" />
    <!--Additional logger elements defined here....-->
  </loggers>
</logging_configuration>
```

When configuring a logger to write messages to either the console or a file, make configuration changes to both the logger and the handler. Setting the level attribute for the logger configures the amount of detail (and therefore, the volume of messages) that the logger sends to the handler. Similarly, setting the level attribute for the handler configures the amount of detail that the handler accepts from the logger.

> **Note:** If you are not getting the volume of output that you expect in a log, then verify that the level attribute for both the logger and the log handler are set appropriately. For example, if the logger is set to TRACE and the log handler is set to WARN, then the handler does not generate messages more detailed than WARN.

8.1.3 Configuring Log Handlers

Individual log handlers are configured in the <log_handlers> section of the logging.xml file. Configure the level attribute for the handler to set the amount of detail that the handler will accept from loggers.

To configure the log handler-level attribute:

> **Note:** You must have a basic understanding of XML syntax before you attempt to modify the logging.xml file.

1. Open the *DOMAIN_NAME*/config/fmwconfig/servers/*SERVER_NAME*/logging.xml file.
2. Change the level attribute as shown in the following examples.

   In this example XML code, the level attribute for the console-handler is set to WARNING:32.

   ```
   <log_handler name='console-handler'
   class='oracle.core.ojdl.logging.ConsoleHandler'
   formatter='oracle.core.ojdl.weblogic.ConsoleFormatter' level='WARNING:32'/>
   ```

For the console-handler to be able to write TRACE level messages to the console, change the level attribute as shown:

```
<log_handler name='console-handler'
class='oracle.core.ojdl.logging.ConsoleHandler'
formatter='oracle.core.ojdl.weblogic.ConsoleFormatter' level='TRACE:1'/>
```

3. Save your changes and restart the application server.

8.1.3.1 Log Handler Configuration Tools

Log handlers that write to a file have additional properties that can be configured. For example, this excerpt from logging.xml configures the odl-handler:

```
<log_handler name='odl-handler' class='oracle.core.ojdl.logging.ODLHandlerFactory'
    filter='oracle.dfw.incident.IncidentDetectionLogFilter'>
    <property name='path'
value='${domain.home}/servers/${weblogic.Name}/logs/${weblogic.Name}-diagnostic.lo
g'/>
    <property name='maxFileSize' value='10485760'/>
    <property name='maxLogSize' value='104857600'/>
    <property name='encoding' value='UTF-8'/>
    <property name='useThreadName' value='true'/>
    <property name='supplementalAttributes' value='J2EE_APP.name,J2EE_MODULE.name,

WEBSERVICE.name,WEBSERVICE_PORT.name,composite_instance_id,component_instance_id,
     composite_name,component_name'/>
</log_handler>
```

To make changes to log handler properties, you can use either the Fusion Middleware Control tool or the WLST command-line tool.

See Also:

- "Configuring Settings for Log Files" in the *Oracle Fusion Middleware Administrator's Guide* for information about both the Fusion Middleware Control tool and the WLST command-line tool
- "Logging Custom WLST Commands" in the *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference* for information about the WLST command-line tool

8.1.4 Configuring Loggers

Individual loggers are configured in the <loggers> section of the logging.xml file. More than twenty different Oracle Identity Manager loggers that can be configured to send messages to log handlers. Oracle Identity Manager loggers are described in Table 2 on page 7.

Setting the level attribute for the logger configures the amount of detail (and, hence, the volume of messages) that the logger sends to its handlers. Nesting one or more <handler> elements inside of <logger> elements assigns handlers to loggers.

The following excerpt shows a logger called OIMCP.PSFTCOMMON. The level attribute is set to WARNING:32 and the logger sends messages to three handlers:

```
<logger name="OIMCP.PSFTCOMMON" level="WARNING:32" useParentHandlers="false">
<handler name="odl-handler"/>
<handler name="wls-domain"/>
<handler name="console-handler"/>
</logger>
```

A logger can inherit a parent logger's settings, including the parent's level setting and other attributes, as well as the parent logger's handlers. To disable inheritance, set the useParentHandlers attribute to false, as shown in the previous excerpt.

At the top of the logger inheritance tree is the root logger. The root logger is the logger with an empty name attribute, as shown in the following example.

```
<loggers>
    <logger name="" level="WARNING:1">
      <handler name="odl-handler"/>
      <handler name="wls-domain"/>
      <handler name="console-handler"/>
    </logger>

    <!-- Additional loggers listed here -->
</loggers>
```

If a logger is configured with only its name attribute, the logger will inherit the rest of its attributes from the root logger, as shown in the following example:

```
<loggers>
    <logger name="oracle.iam.identity.rolemgmt"/>
    <!-- Additional loggers listed here -->
</loggers>
```

To configure loggers:

1. Open the *DOMAIN_NAME*/config/fmwconfig/servers/*SERVER_NAME*/logging.xml file.
2. Locate the logger you want to configure. Table 8–2 lists the Oracle Identity Manager loggers.

Table 8–2 Oracle Identity Manager Loggers

Logger	Description

`oracle.iam.request`
`oracle.iam.requestdatasetgeneration`
`oracle.iam.requestactions`
`oracle.iam.platform.workflowservice`
	Logs events related to request and request dataset management.

`oracle.iam.requesttemplate`	Logs events related to request template management.
`oracle.iam.selfservice`	Logs events related to authenticated and unauthenticated self-service operations.
`oracle.iam.ChangePasswordtaskflow`	Logs events for the password change functionality UI.
`oracle.iam.forgotpasswordtaskflow`	Logs events for the "forgot password" functionality UI.
`oracle.iam.identitytaskflow`	Logs events for the administrative UI identity operations.
`oracle.iam.identity.orgmgmt`	Logs events related to the organization manager service operations.
`oracle.iam.identity.rolemgmt`	Logs events related to the role manager service operations.

Table 8–2 (Cont.) Oracle Identity Manager Loggers

Logger	Description
`oracle.iam.identity.usermgmt`	Logs events related to the user manager service operations.
`oracle.iam.identity.scheduledtasks`	Logs events related to scheduled tasks in the identity feature.
`oracle.iam.platform.utils`	Logs events related to utilities provided by the platform (mainly used by other features). Includes utilities for message resources handling, logging handling, internationalization, caching, and so on.
`oracle.iam.platformservice`	Logs events related to utilities that are mainly executed from the client side. For example, the plug-in registration utility, the purge cache utility, and so on. Some server-side utilities, such as the date-time utility and the exception handling utility, also use this logger.
`oracle.iam.platform.canonic`	Logs events related to the platform UI framework.
`oracle.iam.consoles.faces` `oracle.iam.consoles.common`	Logs messages generated from the UI framework.
`oracle.iam.platform.kernel`	Logs events related to the kernel. This includes the logging generated during the handling of orchestrations by the platform. The event handlers executed in the orchestrations within each feature use that feature's respective logger.
`oracle.iam.platform.context`	Logs events related to the context management feature.
`oracle.iam.platform.entitymgr`	Logs events related to the entity manager feature. This feature provides generic handling of different types of entities, such as users, roles, and so on, and appropriate routing to the respective operations on them.

`oracle.iam.scheduler`
`oracle.iam.platform.scheduler`
`Xellerate.Scheduler`
`Xellerate.Scheduler.Task`
	Logs events related to the scheduler. Note that certain scheduled tasks may also use other loggers.

`oracle.iam.reconciliation`	Logs events related to the reconciliation feature.
`oracle.iam.accesspolicy`	Logs events related to the access policy feature.
`oracle.iam.autoroles`	Logs events related to the auto role membership assignment feature.
`oracle.iam.callbacks`	Logs events related to the callbacks feature.
`oracle.iam.configservice`	Logs events related to the Configuration service APIs that are used for configuration of entity attributes.
`oracle.iam.ldap-sync`	Logs events related to the Oracle Identity Manager and LDAP synchronization feature.
`oracle.iam.notification`	Logs events related to e-mail templates and the notifications handling feature.

Table 8–2 (Cont.) Oracle Identity Manager Loggers

Logger	Description
`oracle.iam.passwdmgnt`	Logs events related to the password management feature.
`oracle.iam.platform.pluginframework`	Logs events from the plug-in framework feature that handles the management of plug-ins.
`oracle.iam.platform.async`	Logs events from platform that handles asynchronous operations.
`oracle.iam.diagnostic`	Logs messages from the diagnostic service APIs used to run diagnostic checks.
`oracle.iam.oimdataproviders`	Logs events related to the Oracle Identity Manager data providers. The Oracle Identity Manager data providers provide code to update and fetch data from the Oracle Identity Manager database.
`Xellerate.Database`	Logs database operations.
`Xellerate.PreparedStatement`	Same as Xellerate.Database, but logs only PreparedStatement details.
`Xellerate.Performance`	Logs database performance, such as time to execute a statement (query), or time to iterate through a result set to get data/metadata.
`oracle.iam.platform.auth`	Logs events for the authentication handling feature.
`oracle.iam.platform.authz` `oracle.iam.authzpolicydefn`	Logs events for the feature that handles authorization policies.
`oracle.iam.sod` `Xellerate.SoD`	Logs events related to SoD (Segregation of Duties).
`oracle.jps`	Logger for the embedded Oracle Entitlements Server MicroSM engine. Note that the log file is created in the *OIM_ORACLE_HOME* folder named as Managed Server name-microsm.log (for example, OIMServer1-microsm.log).
`Xellerate.Entitlement`	Provides logging for entitlement operations used for provisioning entitlements.
`oracle.iam.conf`	Logs events related to the system configuration services feature that includes handling system properties.
`oracle.iam.transUI`	Logs events related to the transitional UI feature that handles initiation of legacy APIs from the 11g code. This includes operations such as initiation of provisioning during user creation, and so on.
`Xellerate.AccountManagement`	Provides logging in legacy user operations APIs.
`Xellerate.Server`	Provides logging in data objects.
`Xellerate.ResourceManagement` `Xellerate.ObjectManagement`	Provides logging for resource object operations.

Table 8–2 (Cont.) Oracle Identity Manager Loggers

Logger	Description
`Xellerate.Workflow`	Provides logging for provisioning process operations.
`Xellerate.WebApp`	Provides logging for the transitional UI operations.
`Xellerate.Adapters`	Provides logging for the adapter factory.
`Xellerate.JavaClient`	Provides logging for client-side data objects.
`Xellerate.Policies`	Provides logging for data objects related to access policies.
`Xellerate.Rules`	Provides logging for data objects related to rules.
`Xellerate.APIs`	Provides logging for legacy public APIs.
`Xellerate.JMS`	Provides logging for JMS operations where messages are produced.
`Xellerate.RemoteManager`	Provides logging in remote manager.
`Xellerate.Auditor`	Provides logging in audit framework.
`Xellerate.Attestation`	Provides logging in the attestation UI and operations.

`Xellerate.GC.StartUp`
`Xellerate.GC.ProviderRegistration`
`Xellerate.GC.ImageGeneration`
`Xellerate.GC.FrameworkProvisioning`
`Xellerate.GC.Provider.ProvisioningFormat`
`Xellerate.GC.Provider.ProvisioningTransport`
`Xellerate.GC.FrameworkReconciliation`
`Xellerate.GC.Provider.ReconciliationFormat`
`Xellerate.GC.Provider.Validation`
`Xellerate.GC.Provider.Transformation`
`Xellerate.GC.Model`
`Xellerate.GC.Server`
	Provides logging for the Generic Technology Connector (GTC).

3. Define the level attribute for the <logger> element. See the example at the beginning of this section.
4. Add one or more <handler> elements to the <logger> element.
5. When you are finished editing both the <loggers> and <log_handlers> sections of logging.xml, save the file.
6. Restart the application server for the changes to take effect.

8.1.5 Sample ODL Log Output

The following ODL log excerpt illustrates the kind of output you can expect.

```
<Jun 15, 2010 2:01:20 AM IST> <Error> <oracle.iam.platform.authz.impl>
<IAM-1010032>
<No OES Policy found for the given Action.>
<Jun 15, 2010 2:02:02 AM IST> <Warning> <oracle.iam.platform.canonic.agentry>
<IAM-0091108> <readme.txt is not a valid connector resource file.>
<Jun 15, 2010 2:02:52 AM IST> <Error> <oracle.iam.configservice.impl>
<IAM-3020003> <The attribute User Type does not exist!>
```

For information about managing and interpreting log output, see "Managing Log Files and Diagnostic Data" in the *Oracle Fusion Middleware Administrator's Guide*.

8.2 Logging in Oracle Identity Manager By Using log4j

Apache log4j is used with third-party applications, such as Nexaweb for Deployment Manager and Workflow Designer, and OSCache for caching.

The location of the log4j configuration file is:

OIM_HOME/config/log.properties

Logging in Oracle Identity Manager by using log4j is described in the following sections:

- Log Levels
- Loggers
- Configuring and Enabling Logging

8.2.1 Log Levels

Table 8–3 lists the log levels for log4j:

Table 8–3 Log Levels for log4j

Log Level	Description
DEBUG	The DEBUG level designates fine-grained informational events that are useful to debug an application.
INFO	The INFO level designates informational messages that highlight the progress of the application at coarse-grained level.
WARN	The WARN level designates potentially harmful situations.
ERROR	The ERROR level designates error events that might allow the application to continue running.
ALL	The ALL level has the lowest possible rank and is intended to turn on all logging.
OFF	The OFF level has the highest possible rank and is intended to turn off logging.

8.2.2 Loggers

The loggers for the third-party applications used are:

- com.nexaweb.server for Nexaweb
- com.opensymphony.oscache for OSCache

8.2.3 Configuring and Enabling Logging

Any of the log levels can be used for the third-party applications as follows:

```
log4j.logger.com.nexaweb.server=WARN
log4j.logger.com.opensymphony.oscache=ERROR
```

9

Enabling Secure Cookies

By default, Oracle Identity Manager can be accessed over HTTP but does not work over Secure Socket Layer (SSL). This is because the cookie-secure flag is disabled by default. The cookie-secure flag tells the Web browser to only send the cookie back over an HTTPS connection. This ensures that the cookie is transmitted only on a secure channel. HTTPS must be enabled for the URL exposed by the application.

To enable Oracle Identity Manager to work over SSL, you must enable the cookie-secure flag. To do so:

1. Add the <cookie-secure>true</cookie-secure> tag to the following files in the Oracle Identity Manager deployment:
   - *OIM_HOME*/apps/oim.ear/admin.war/WEB-INF/weblogic.xml
   - *OIM_HOME*/apps/oim.ear/iam-consoles-faces.war/WEB-INF/weblogic.xml
   - *OIM_HOME*/apps/oim.ear/xlWebApp.war/WEB-INF/weblogic.xml
2. Create a new weblogic.xml file for Nexaweb application if it does not exist in its WEB-INF/ directory.
3. Add the following session descriptor in it:

```
.
OIM_HOME/apps/Nexaweb.ear/Nexaweb.war/WEB-INF/weblogic.xml
.
.
<session-descriptor>

<persistent-store-type>replicated_if_clustered</persistent-store-type>
             <cookie-http-only>false</cookie-http-only>
             <cookie-name>oimjsessionid</cookie-name>
                          <cookie-secure>true</cookie-secure>
     <url-rewriting-enabled>false</url-rewriting-enabled>
 </session-descriptor>
```

4. Save weblogic.xml.

10

Configuring LDAP Authentication When LDAP Synchronization is Enabled

Use the following procedure to be able to use LDAP for authentication when LDAP synchronization is enabled.

Note: This procedure does not enable the following functionality:

- Forced password changes, including first login, administrator password reset, and expired passwords
- Forced setting of challenge responses

1. Add a dynamic group in Oracle Internet Directory (OID).
   a. Create an oimusers.ldif file that defines a dynamic group. The format of the LDIF file should be similar to the following:

```
dn: cn=oimusers, <group search base>
objectclass: orclDynamicGroup
objectclass: groupOfUniqueNames

labelederuri:ldap://<LdapHost>:<LdapPort>/<UserSearchBase>??sub?(objectclass=
inetOrgPerson)
```

      For example:

```
dn: cn=oimusers,cn=Groups,dc=us,dc=oracle,dc=com
objectclass: orclDynamicGroup
objectclass: groupOfUniqueNames
labelederuri:
ldap://dadvmc0225:3060/cn=Users,dc=us,dc=oracle,dc=com??sub?(objectclass=in
etOrgPerson)
```

   b. Use the ldapadd command to upload the oimusers.ldif file to OID. The command should have the following format:

```
ldapadd -h <ldaphost> -p <ldapport> -D <root dn> -w <password> -f
oimusers.ldif
```

      For example:

```
ldapadd -h dadvmc0225 -p 3060 -D cn=orcladmin -w welcome1 -f oimusers.ldif
```

   c. Use the ldapsearch command to validate group members. The command should have the following format:

```
ldapsearch -h <ldaphost> -p <ldapport> -D <root dn> -w <password> -b
"cn=oimusers,<groupsearchbase>" -s base "objectclass=*"
```

For example:

```
ldapsearch  -h dadvmc0225 -p 3060 -Dcn=orcladmin -wwelcome1 -b
"cn=oimusers,cn=Groups,dc=us,dc=oracle,dc=com" -s base "objectclass=*"
```

2. Configure the LDAP Authenticator in WLS.
   a. Log in to WebLogic Administrative Console.
   b. Go to Security Realms, myrealm, Providers.
   c. Click **New**. Give a name and choose OracleInternetDirectoryAuthenticator as type.
   d. Set the Control Flag to SUFFICIENT.
   e. Click the Provider Specific settings and configure the OID connection details.
   f. In Dynamic groups section, enter the following values:

      Dynamic Group Name Attribute: cn

      Dynamic Group Object Class: orcldynamicgroup

      Dynamic Member URL Attribute: labeleduri

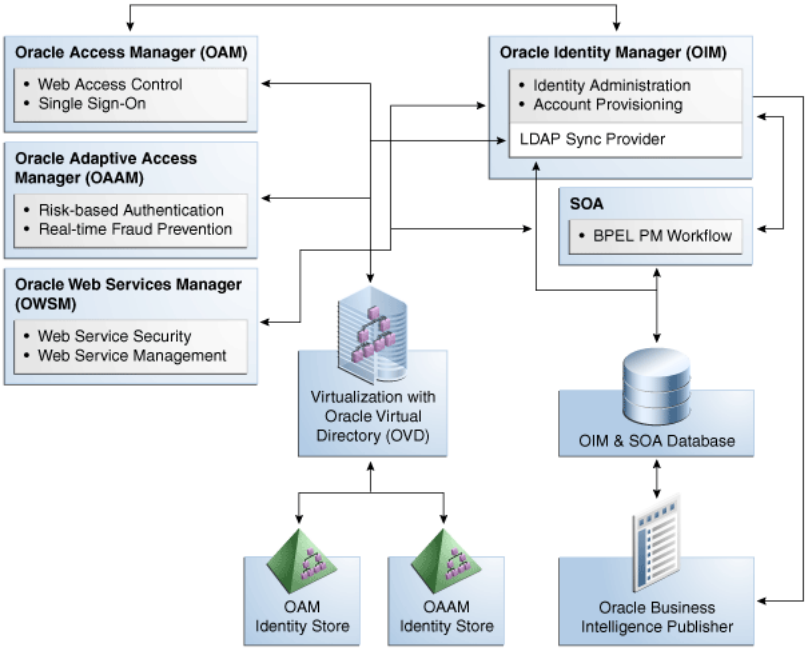
      User Dynamic Group DN Attribute: GroupOfUniqueNames
   g. Click the Providers tab and then click **Reorder**. Reorder the LDAP authenticator so this is placed before the OIM Authenticator.
3. Restart all servers.
4. Validate role memberships.
   a. Login to WebLogic Admin Console.
   b. Go to Security Realms, myrealm, User and Groups.
   c. Click **users** to display all the users in the LDAP user search base. If the LDAP users are not displayed, it means that there is an error with the LDAP connection, and the details are specified in OID Authenticator (provider specific settings).
   d. Click on any user and then to the corresponding group entry. "Oimusers" should be one of the listed entries. If this validation fails, please go through the LDAP authenticator's provider-specific details.

11

Integrating with Other Oracle Components

Oracle offers several technologies that compliment and extend the functionality available in Oracle Identity Manager, some of which are described in this chapter. Refer to the "Oracle Fusion Middleware Integration Overview" for complete information about the technologies you can integrate with Oracle Identity Manager. Figure 11–1 shows the integration of Oracle Identity Manager with other Oracle components.

Figure 11–1 Integration with Other Components



This chapter discusses the integration of Oracle Identity Manager with the following Oracle components:

- Oracle Access Manager
- Oracle Adaptive Access Manager
- Oracle Identity Analytics
- Oracle Identity Navigator
- Oracle Virtual Directory
- Oracle Service-Oriented Architecture

- Oracle Business Intelligence Publisher

11.1 Oracle Access Manager

Oracle Access Manager (OAM) protects applications, data, and cloud-based services through a combination of flexible authentication and single sign-on (SSO), identity federation, risk-based authentication, proactive enterprise fraud prevention, and fine-grained authorization.

Web-based SSO provides secure access to multiple applications with one authentication step. When OAM is combined with Oracle Identity Manager, OAM can SSO-enable the Oracle Identity Administration, along with the other Oracle Identity Management components.

Oracle Identity Manager, OAM, and Oracle Adaptive Access Manager (OAAM) share a common set of LDAP attributes, improving efficiency by making it easier to manage workflows and other processes. Integrated password management makes it easy for users to log in to OAM, OAAM, and Oracle Identity Manager, and to manage expired and forgotten passwords.

For integration details, see "Integration Between OIM and OAM" in the *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

11.2 Oracle Adaptive Access Manager

OAAM provides sophisticated multifactor authentication and proactive, real-time fraud prevention functionality for Web-based connections.

Risk-based authentication is one such capability OAAM provides. The OAAM risk-scoring engine combats identity fraud in real-time by evaluating whether a user should be allowed to authenticate based on the type of transaction being attempted and the probability of fraud occurring. Next, the OAAM risk-scoring engine evaluates how a user answers a series of dynamically generated questions that are created based on a combination of public and private data sources. OAAM then generates a fraud score and the user is either allowed to continue with the transaction or is denied access.

When integrated with Oracle Identity Manager, the robust challenge question feature set found in OAAM replaces the more limited set found in Oracle Identity Manager, which handles password validation, storage, and propagation duties.

For information about how password management is achieved when Oracle Identity Manager is integrated with OAM and OAAM, see "Deployment Options for Password Management" in the *Oracle Fusion Middleware Integration Guide for Oracle Access Manager.*

For integration details, see "Integrating Oracle Access Manager, Oracle Adaptive Access Manager, and Oracle Identity Manager" in the *Oracle Fusion Middleware Integration Guide for Oracle Access Manager.*

11.3 Oracle Identity Analytics

Oracle Identity Analytics (OIA), formerly Sun Role Manager, provides rich identity analytics and dashboards that allow you to monitor, analyze, review, and govern user access in order to mitigate risk, build transparency, and satisfy compliance mandates.

When integrated with Oracle Identity Manager, Oracle Identity Analytics defines the Role-based Access Control (RBAC) framework, the attestation process, and the

approach to Segregation of Duties (SoD) policy enforcement, while Oracle Identity Manager serves as the automated provisioning and identity synchronization solution. Rather than assigning individual access entitlements, the RBAC framework allows organizations to assign and unassign roles as a means of controlling user access on various applications.

For integration details, see "Integrating With Oracle Identity Manager" in the *Oracle Identity Analytics 11gR1 System Integrator's Guide*. You can access the Oracle Identity Analytics 11gR1 System Integrator's Guide at the following URL:

`http://wikis.sun.com/x/UBbeCw`

11.4 Oracle Identity Navigator

Oracle Identity Navigator (OIN) is a browser-based administrative portal designed to act as a launch pad for Oracle Identity Management components. It does not replace the individual component consoles. Rather, it allows you to access the Oracle Identity Management consoles from one site.

When integrated with Oracle Identity Manager, OIN replaces the Oracle Identity Administration as the primary Oracle Identity Manager user interface.

OIN has a product discovery feature that can be used to discover all active J2EE components in a domain, including the Oracle Identity Administration.

For integration details, see "Adding a Component Link to the Product Launcher by Using Product Discovery" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Navigator*.

11.5 Oracle Virtual Directory

If you install Oracle Identity Manager with LDAP, you must install Oracle Virtual Directory (OVD). OVD connects to multiple enterprise directories and consolidates the contents of those directories into a unified view. For example, if your enterprise uses Oracle Internet Directory (OID), iPlanet, and Active Directory, OVD can interface with all three directories and create a consolidated view. Oracle Identity Manager can then use a single connector to access the consolidated LDAP data on OVD. The LDAP Sync Provider (also called the LDAP Provider) connects Oracle Identity Manager and OVD.

When integrated with Oracle Identity Manager, OVD provides the following benefits:

- Oracle Identity Manager connector management is simplified - Only a single LDAP connector is needed for multiple directory providers (although, multiple instances may be needed)
- LDAP connector reliability is improved - The same connector is used regardless of the underlying LDAP server. OVD handles the data translation that, in the past, required multiple LDAP connectors for multiple LDAP providers
- The same identity virtualization capability is provided to all Fusion Middleware applications, reducing the overall footprint of components in the Enterprise

For integration details, see the *"Oracle Fusion Middleware Installation Guide for Oracle Identity Management*", which contains multiple procedures for integrating Oracle Identity Manager and Oracle Virtual Directory in various environments.

11.6 Oracle Service-Oriented Architecture

The Oracle Identity Manager workflow feature utilizes Oracle Service-Oriented Architecture (SOA) back-end services and management capabilities to provide an interactive environment to request, approve, and manage user access. In order to install Oracle Identity Manager, you also must install Oracle SOA.

Oracle Identity Manager makes use of the following SOA Suite components:

- BPEL Process Manager, which provides the end-to-end solution for creating and managing business processes
- Human Workflow, which manages the lifecycle of human tasks, including creation, assignment, deadlines, expiration, and notifications
- Oracle Business Rules, which allows you to define complex business rules to support request assignment, process selection, and approver resolution
- Oracle Web Services Manager, which secures the web service and BPEL processes consumed and invoked by Oracle Identity Manager

For integration details, see "Integration with Oracle SOA Suite" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

11.7 Oracle Business Intelligence Publisher

The Oracle Identity Manager reporting feature utilizes Oracle Business Intelligence Publisher (BI Publisher) to provide high-fidelity reporting capabilities, allowing you to create, deploy, and use complex reports in a multi-channel environment.

For BI Publisher details, see "Using Reporting Features" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager.*

12

Handling Lifecycle Management Changes

Because of integrated deployment of Oracle Identity Manager with other applications, such as Oracle Access Manager (OAM), and configuration changes in those applications, various configuration changes might be required in Oracle Identity Manager and Oracle WebLogic Server. These configuration changes are described in the following sections:

- URL Changes Related to Oracle Identity Manager
- Password Changes Related to Oracle Identity Manager
- Configuring SSL for Oracle Identity Manager

12.1 URL Changes Related to Oracle Identity Manager

Oracle Identity Manger uses various hostname and port in its configuration because of the architectural and middleware requirements. This section describes ways to make the corresponding changes in Oracle Identity Manager and Oracle WebLogic configuration for any change in the integrated and dependent applications.

This section contains the following topics:

- Oracle Identity Manager Database Host and Port Changes
- Oracle Virtual Directory Host and Port Changes
- Oracle Identity Manager Host and Port Changes
- BI Publisher Host and Port Changes
- SOA Host and Port Changes
- OAM Host and Port Changes

12.1.1 Oracle Identity Manager Database Host and Port Changes

This section describes the configuration areas where database hostname and port number are used.

After installing Oracle Identity Manager, if there are any changes in the database hostname or port number, then the following changes are required:

> **Note:** Before making changes to the database host and port, shutdown the managed servers hosting Oracle Identity Manager. But you can keep the Oracle WebLogic Administrative Server running.

- **To change datasource oimJMSStoreDS configuration:**

1. Navigate to **Services**, **JDBC**, **Data Sources**, and then **oimJMSStoreDS**.
2. Click the **Connection Pool** tab.
3. Modify the values of the **URL** and **Properties** fields to reflect the changes to database host and port.

- **To change datasource oimOperationsDB configuration:**
  1. Navigate to **Services**, **JDBC**, **Data Sources**, and then **oimJMSStoreDS**.
  2. Click the **Connection Pool** tab.
  3. Modify the values of the URL and Properties fields to reflect the changes to database host and port.
- **To change the datasource related to Oracle Identity Manager Meta Data Store (MDS) configuration:**
  1. Navigate to **Services**, **JDBC**, **Data Sources**, and then **mds-oim**.
  2. Click the **Connection Pool** tab.
  3. Modify the values of the URL and Properties fields to reflect the changes in the database host and port.
- **To change OIMAuthenticationProvider configuration:**
  1. In the WebLogic Administrative console, navigate to **Security Realms**, **myrealm**, and then **Providers**.
  2. Click **OIMAuthenticationProvider**.
  3. Click **Provider Specific**.
  4. Modify the value of the DBUrl field to reflect the change in hostname and port.

     Note: If Service Oriented Architecture (SOA) and Oracle Web Services Manager (OWSM) undergo configuration changes, then you must make similar changes for datasources related to SOA or OWSM.

  After making changes in the datasources, restart the Oracle WebLogic Administrative Server, and start the Oracle Identity Manager managed WebLogic servers.
- **To change DirectDB configuration:**
  1. Login to Enterprise Manager by using the following URL:

     http://*ADMIN_SERVER*/em
  2. Navigate to **Identity and Access**, and then **oim**.
  3. Right-click **oim**, and navigate to **System MBean Browser** under Application Defined MBeans.
  4. Navigate to **oracle.iam**, **Application:oim**, **XMLConfig**, **Config**, **XMLConfig.DirectDBConfig**, and then **DirectDB**.
  5. Enter the new value for the URL attribute to reflect the changes to host and port, and then apply the changes.

Note: When Oracle Identity Manager single instance deployment is changed to Oracle Real Application Clusters (Oracle RAC) or Oracle RAC is changed to single instance deployment, change the oimJMSStoreDS, oimOperationsDB, and mds-oim datasources. In addition to the generic changes to make these datasources to multidatasource configuration, change the OIMAuthenticationProvider and domain credential store configurations to reflect the Oracle RAC URL.

12.1.2 Oracle Virtual Directory Host and Port Changes

When LDAP synchronization is enabled, Oracle Identity Manager connects with directory servers through Oracle Virtual Directory (OVD). This connection takes place by using LDAP/LDAPS protocol.

To change OVD host and port:

1. Login to Oracle Identity Manager Administration.
2. Click **Advanced**.
3. Under Configuration, click **Manage IT Resource**.
4. From the IT Resource Type list, select **Directory Server** , and click **Search**.
5. Edit the Directory Server IT resource. To do so:
   a. If the value of the Use SSL field is set to `False`, then edit the Server URL field. If the value of the Use SSL field is set to `True`, then edit the Server SSL URL field.
   b. Click **Update**.

12.1.3 Oracle Identity Manager Host and Port Changes

This section consists of the following topics:

- Changing OimFrontEndURL in Oracle Identity Manager Configuration
- Changing backOfficeURL in Oracle Identity Manager Configuration

Note: When additional Oracle Identity Manager nodes are added or removed, perform the procedures described in these sections to configure Oracle Identity Manager host and port changes.

12.1.3.1 Changing OimFrontEndURL in Oracle Identity Manager Configuration

The OimFrontEndURL is the URL used to access the Oracle Identity Manager UI. This can be a load balancer URL or Web server URL depending on the application server is fronted with loan balancer or Web server, or single application server URL. This is used by Oracle Identity Manager in the notification e-mails as well as the callback URL for SOA calls.

The change may be necessary because of change in Web server hostname or port for Oracle Identity Manager deployment in a clustered environment, or WebLogic managed server hostname or port changes for Oracle Identity Manager deployment in a nonclustered environment.

To change the OimFronEndURL in Oracle Identity Manager configuration:

1. Login to Enterprise Manager by using the following URL when the WebLogic Administrative Server and Oracle Identity Manager managed servers, at least one of the servers in case of a clustered deployment, are running:

   http://*ADMIN_SERVER*/em

2. Navigate to **Identity and Access**, **oim**.
3. Right-click **oim**, and navigate to **System MBean Browser**.
4. Under Application Defined MBeans, navigate to **oracle.iam**, **Application:oim**, **XMLConfig**, **Config**, **XMLConfig.DiscoveryConfig**, and then **Discovery**.
5. Enter new value for the OimFrontEndURL attribute, and click **Apply** to save the changes. Example values can be:

   http://myoim.oracle.com

   https://myoim.oracle.com

   http://myserver.oracle.com:7001

   Note: SPML clients store Oracle Identity Manager URL for invoking SPML and sending callback response. Therefore, changes are required corresponding to this. In addition, if Oracle Identity Manager is integrated with OAM, OAAM, or Oracle Identity Navigator (OIN), there may be corresponding changes necessary. For more information, refer to OAM, OAAM, and OIN documentation in the Oracle Technology Network (OTN) Web site.

12.1.3.2 Changing backOfficeURL in Oracle Identity Manager Configuration

Changing backOfficeURL is required only for Oracle Identity Manager deployed in front-office and back-office configuration. This change does not apply for simple clustered or nonclustered deployments. This URL is used internally by Oracle Identity Manager for accessing back-office components from the front-office components. You might change the value of this attribute during the implementation of back-office and front-office configuration, for adding additional servers to back office, and for removing servers from back-office.

To change the value of the backOfficeURL attribute:

1. Login to Enterprise Manager by using the following URL when the WebLogic Administrative Server and Oracle Identity Manager managed servers, at least one of the servers in case of a clustered deployment, are running:

   http://*ADMIN_SERVER*/em

2. Navigate to **Identity and Access**, and then **oim**.
3. Right-click **oim**, and navigate to **System MBean Browser**.
4. Under Application Defined MBeans, navigate to **oracle.iam**, **Application:oim**, **XMLConfig**, **Config**, **XMLConfig.DiscoveryConfig**, **Discovery**.
5. Enter a new value for the BackOfficeURL attribute, and click **Apply** to save the changes. Example values can be:

   t3://mywls1.oracle.com:8001

   t3://mywls1.oracle.com:8001,mywls2.oracle.com:9001

Note: The value of the BackOfficeURL attribute must be empty for Oracle Identity Manager nonclustered and clustered deployments.

12.1.4 BI Publisher Host and Port Changes

BI Publisher can be accessed by clicking a simple link from Oracle Identity Manager Administrative and User console for reporting purposes. This URL is based on the configuration value on Oracle Identity Manager side. If there is host and port changes for BI Publisher, then the following change must be made in Oracle Identity Manager:

1. Login to Enterprise Manager by using the following URL when the WebLogic Administrative Server and Oracle Identity Manager managed servers, at least one of the servers in case of a clustered deployment, are running:

   http://*ADMIN_SERVER*/em

2. Navigate to **Identity and Access**, **oim**.
3. Right-click **oim**, and navigate to **System MBean Browser**.
4. Under Application Defined MBeans, navigate to **oracle.iam**, **Application:oim**, **XMLConfig**, **Config**, **XMLConfig.DiscoveryConfig**, **Discovery**.
5. Enter a new value for the BIPublisherURL attribute, and click **Apply** to save the changes.

12.1.5 SOA Host and Port Changes

To change the SOA host and port:

Note: When additional SOA nodes are added or removed, perform this procedure to change the SOA host and port.

1. Login to Enterprise Manager by using the following URL when the WebLogic Administrative Server and Oracle Identity Manager managed servers, at least one of the servers in case of a clustered deployment, are running:

   http://*ADMIN_SERVER*/em

2. Navigate to **Identity and Access**, **oim**.
3. Right-click **oim**, and navigate to **System MBean Browser**.
4. Under Application Defined MBeans, navigate to **oracle.iam**, **Application:oim**, **XMLConfig**, **Config**, **XMLConfig.SOAConfig**, **SOAConfig**.
5. Change the values of the Rmiurl and Soapurl attributes, and click **Apply** to save the changes.

   The Rmiurl attribute is used for accessing SOA EJBs deployed on SOA managed servers. This is the application server URL. For a clustered deployment of Oracle Identity Manager, it is a comma-separated list of all the SOA managed server URLs. Example values for this attribute can be:

   t3://mysoa1.oracle.com:8001

   t3s://mysoa1.oracle.com:8001,mysoa2.oracle.com:8002

   t3://mysoa1.oracle.com:8001,mysoa2.oracle.com:8002,mysoa3.oracle.com:8003

The Soapurl attribute is used for accessing SOA Web services deployed on SOA managed servers. This is the Web server and load balancer URL for a SOA cluster front-ended with Web server and load balancer. It can be application server URL for a single SOA server.

The example values for this attribute can be:

http://myoimsoa.oracle.com

http://mysoa.oracle.com:8001

12.1.6 OAM Host and Port Changes

To change the OAM host and port:

1. Login to Enterprise Manager by using the following URL when the WebLogic Administrative Server and Oracle Identity Manager managed servers, at least one of the servers in case of a clustered deployment, are running:

   http://*ADMIN_SERVER*/em
2. Navigate to **Identity and Access**, **oim**.
3. Right-click **oim**, and navigate to **System MBean Browser**.
4. Under Application Defined MBeans, navigate to **oracle.mds.lcm**, **Server:oim_server1**, **Application:oim**, **MDSAppRuntime**.
5. Click the **Operations** tab, and then click **exportMetaData**.
6. In the toLocation field, enter `/tmp` or another directory.
7. Select the value of createSubDir as **False**.
8. In the Element field, enter **/db/oim-config.xml**.
9. Select **False** for excludeAllCust, excludeBaseDocs and excludeExtendedMetadata.
10. Click **Invoke**. This exports the oim-config.xml file to the directory specified in the toLocation field.
11. In the oim-config.xml file, modify the ssoConfig element. This element contains host, port, and other configuration properties related to OAM.
12. In the Enterprise Manager console, navigate to **Identity and Access**, **oim**.
13. Right-click **oim**, and navigate to **System MBean Browser**.
14. Under Application Defined MBeans, navigate to **oracle.mds.lcm**, **Server:oim_server1**, **Application:oim**, **MDSAppRuntime**.
15. Click the **Operations** tab, and then click **importMetaData**.
16. In the fromLocation field, enter **/tmp** or another directory.
17. Select the value of createSubDir as **False**.
18. In the Element field, enter **/db/oim-config.xml**.
19. Select **False** for excludeAllCust, excludeBaseDocs and excludeExtendedMetadata.
20. Click **Invoke**. This action imports the file in the Oracle Identity Manager meta data repository.

12.2 Password Changes Related to Oracle Identity Manager

Various passwords are used for Oracle Identity Manger configuration because of the architectural and middleware requirements. This section describes the default passwords and ways to make the changes to the password in Oracle Identity Manger and Oracle WebLogic configuration for any change in the dependent or integrated products.

This section consists of the following topics:

- Changing Oracle WebLogic Administrator Password
- Changing Oracle Identity Manager Administrator Password
- Changing Oracle Identity Manager Database Password
- Changing Oracle Identity Manager Passwords in the Credential Store Framework
- Changing OVD Password

12.2.1 Changing Oracle WebLogic Administrator Password

To change Oracle WebLogic administrator password:

1. Login to WebLogic Administrative console.
2. Navigate to **Security Realms**, **myrealm**, **Users and Groups**, **weblogic**, **Password**.
3. In the New Password field, enter the new password.
4. In the Confirm New Password field, re-enter the new password.
5. Click **Apply**.

12.2.2 Changing Oracle Identity Manager Administrator Password

During Oracle Identity Manager installation, the installer prompts for the Oracle Identity Manager administrator password. If required, you can change the administrator password after the installation is complete. To do so, you must login to Oracle Identity Manager Self Service as Oracle Identity Manager administrator. For information about how to change the administrator password, see "Authenticated User Self Service" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager*.

Note: If OAM or OAAM is integrated with Oracle Identity Manager, then you might have to make corresponding changes in those applications. For more information, refer to OAM and OAAM documentation in the Oracle Technology Network (OTN) Web site by using the following URL:

`http://www.oracle.com/technology/documentation/oim1014.html`

12.2.3 Changing Oracle Identity Manager Database Password

Oracle Identity Manager uses two database schemas for storing Oracle Identity Manager operational and configuration data. It uses Oracle Identity Manager MDS schema for storing configuration-related information and Oracle Identity Manager schema for storing other information. Any change in the schema password requires changes on Oracle Identity Manager configuration.

Changing Oracle Identity Manager database password involves the following:

> **Note:** Before changing the database password, shutdown the managed servers that host Oracle Identity Manager. However, you can keep the Oracle WebLogic Administrative Server running.

- **To change datasource oimJMSStoreDS configuration:**
  1. Navigate to **Services**, **JDBC**, **Data Sources**, **oimJMSStoreDS**.
  2. Click the **Connection Pool** tab.
  3. In the Password and Confirm password fields, enter the new Oracle Identity Manager database schema password.
  4. Click **Save** to save the changes.
- **To change datasource oimOperationsDB configuration:**
  1. Navigate to **Services**, **JDBC**, **Data Sources**, **oimJMSStoreDS**.
  2. Click the **Connection Pool** tab.
  3. In the Password and Confirm password fields, enter the new Oracle Identity Manager database schema password.
  4. Click **Save** to save the changes.
- **To change datasource related to Oracle Identity Manager MDS configuration:**
  1. Navigate to **Services**, **JDBC**, **Data Sources**, **mds-oim**.
  2. Click the **Connection Pool** tab.
  3. In the Password and Confirm password fields, enter the new Oracle Identity Manager MDS database schema password.
  4. Click **Save** to save the changes.

  > **Note:**
  >
  > - For Oracle Identity Manager deployments with Oracle Real Application Clusters (Oracle RAC) configuration, you might have to make changes in all the datasources under the respective multi-datasource configurations.
  > - You might have to make similar changes for datasources related to SOA or OWSM, if required.

- **To change OIMAuthenticationProvider configuration:**
  1. In the WebLogic Administrative console, navigate to **Security Realms**, **myrealm**, and then **Providers**.
  2. Click **OIMAuthenticationProvider**.
  3. Click **Provider Specific**.
  4. In the DBPassword field, enter the new Oracle Identity Manager database schema password.
  5. Click **Save** to save the changes.
- **To change domain credential store configuration:**

1. Login to Enterprise Manager by using the following URL:

   http://*ADMIN_SERVER*/em
2. Navigate to **Weblogic Domain**, and then ***DOMAIN_NAME***.
3. Right click **oim**, and navigate to **Security**, **Credentials**, and then **oim**.
4. Select **OIMSchemaPassword**, and click **Edit**.
5. In the Password field, enter the new password, and click **OK**.

After changing the Oracle Identity Manager database password, restart the WebLogic Administrative Server. Start the Oracle Identity manager managed WebLogic Servers as well.

12.2.4 Changing Oracle Identity Manager Passwords in the Credential Store Framework

Oracle Identity Manager installer stores several passwords during the install process. Various values are stored in Credential Store Framework (CSF) as key and value. Table 12–1 lists the keys and the corresponding values:

Table 12–1 CSF Keys

Key	Description
DataBaseKey	Password for the key used for encrypt database. The default password generated by Oracle Identity Manager installer is xellerate.
.xldatabasekey	Password for keystore that stores the database encryption key. Password is user input value in installer for the OIM Keystore Password field.
xell	This is the password for key 'xell', which is used for securing communication between Oracle Identity Manager components. Default password generated by Oracle Identity Manager installer is xellerate.
default_keystore.jks	This is the password for the default_keystore.jks JKS keystore in the *DOMAIN_HOME*/config/fmwconfig/ directory.
SOAAdminPassword	Password is user input value in the installer for SOA Administrator Password field.
OIMSchemaPassword	This is the password for connecting to Oracle Identity Manager database schema. Password is user input value in the installer for OIM Database Schema Password field.
JMSKey	Default password generated by Oracle Identity Manager installer is xellerate.

To change the values of the CSF keys:

1. Login to Enterprise Manager.
2. Right-click the domain.
3. Navigate to **Security**, and then **Credential**.
4. Expand **oim**. The list of all the key and value pairs for Oracle Identity Manager are displayed. You can edit and change the values.

12.2.5 Changing OVD Password

To change the OVD password:

1. Login to Oracle Identity Manager Administration.
2. Click **Advanced**.
3. Under Configuration, click **Manage IT Resource**.
4. From the IT Resource Type list, select **Directory Server**.
5. Click **Search**.
6. Edit the Directory Server IT resource. To do so, in the Admin Password field, enter the new OVD password, and click **Update**.

12.3 Configuring SSL for Oracle Identity Manager

This section explains the procedure for setting up SSL Configuration for Oracle Identity Manager and for the components with which Oracle Identity Manager interacts and establish secure communication between them. It includes the following topics:

- Enabling SSL for Oracle Identity Manager and SOA Servers
- Enabling SSL for Oracle Identity Manager DB
- Enabling SSL for LDAP Synchronization

12.3.1 Enabling SSL for Oracle Identity Manager and SOA Servers

You need to perform the following configurations in Oracle Identity Manager and SOA servers to enable SSL:

- Enabling SSL for Oracle Identity Manager and SOA WebLogic Server
- Changing OimFrontEndURL to Use SSL Port
- Changing backOfficeURL to Use SSL Port
- Changing SOA Server URL to Use SSL Port
- Configuring SSL for Design Console
- Configuring SSL for Oracle Identity Manager Utilities
- Configuring SSL for MDS Utilities
- Configuring SSL for SPML/Callback Domain

12.3.1.1 Enabling SSL for Oracle Identity Manager and SOA WebLogic Server

To enable SSL for Oracle Identity Manager and SOA servers:

1. Log in to WebLogic Server Administrative console and go to Servers, OIM_SERVER1, General. Under the general section, you can enable ssl port to any value and activate it.
2. The server will start listening and you can access the URL with HTTPS protocol.
3. Perform the same steps for Admin/SOA Servers as Oracle Identity Manager might need to interact with SSL-enabled SOA Server.

After enabling SSL on Oracle Identity Manager and SOA Servers, perform the following changes for establishing secured communication between them:

- Changing OimFrontEndURL to Use SSL Port
- Changing backOfficeURL to Use SSL Port

- Changing SOA Server URL to Use SSL Port

12.3.1.2 Changing OimFrontEndURL to Use SSL Port

OimFrontEndURL is used to access the oim application UI. This can be a load balancer URL or web server URL (in case application server is fronted with load balancer or web server) or single application server URL. This is generally used by Oracle Identity Manager in the notificaiton emails or to send a call back web service from SOA to Oracle Identity Manager.

To change the OimFrontEndURL to use SSL port:

1. When the WebLogic admin and Oracle Identity Manager managed servers (at least one of the servers in case of cluster) are running, log in to Enterprise Manager (EM).

   For example:

   http://<AdminServer>/em

2. Navigate to Identity and Access, Oracle Identity Manager.
3. Right click and select System MBean Browser.
4. Under Application Defined MBeans, navigate to oracle.iam, Application:oim, XMLConfig, Config, XMLConfig.DiscoveryConfig, Discovery.
5. Enter a new value for the "OimFrontEndURL" attribute and click **Apply** to save the changes.

   For example:

   http://myoim.oracle.com

   https://myoim.oracle.com

   http://myserver.oracle.com:7001

> **Note:** SPML clients store Oracle Identity Manager URL for invoking SPML and also send callback response. Therefore, there will be changes needed corresponding to this. Also, if Oracle Identity Manager is integrated with OAM/OAAM/OIN, there may be corresponding changes necessary. Refer to Chapter 11, "Integrating with Other Oracle Components" for detailed information about the integration with other components.

12.3.1.3 Changing backOfficeURL to Use SSL Port

backOfficeURL change is required only for Oracle Identity Manager deployed in front-office/back-office configuration. For simple cluster or non-cluster installations the following does not apply. This URL is used internally by Oracle Identity Manager for accessing back-office components from the front-office components. This value needs to be changed initially during the implementation of back-office/front-office configuration, for adding additional servers to back office, and for removing servers from back-office.

To change the backOfficeURL to use SSL port:

1. When the WebLogic admin and Oracle Identity Manager managed servers (at least one of the servers in case of cluster) are running, log in to Enterprise Manager (EM).

   For example:

http://<AdminServer>/em

2. Navigate to Identity and Access, Oracle Identity Manager.
3. Right click and select System MBean Browser.
4. Under Application Defined MBeans, navigate to oracle.iam, Application:oim, XMLConfig, Config, XMLConfig.DiscoveryConfig, Discovery.
5. Enter a new value for the "backOfficeURL" attribute and click **Apply** to save the changes.

   For example:

   t3://mywls1.oracle.com:8001

   t3://mywls1.oracle.com:8001,mywls2.oracle.com:9001

   Note: For simple cluster and non-cluster installations the value must be empty.

12.3.1.4 Changing SOA Server URL to Use SSL Port

To change SOA server URL to use SSL port:

1. When the admin server and Oracle Identity Manager managed servers are running, log in to Enterprise Manager (EM).

   For example:

   http://<AdminServer>/em

2. Navigate to Identity and Access, Oracle Identity Manager.
3. Right click and select System MBean Browser.
4. Under Application Defined MBeans, navigate to oracle.iam, Application:oim, XMLConfig, Config, XMLConfig.SOAConfig, SOAConfig.
5. Change the values for attributes "Rmiurl", "Soapurl", and click **Apply** to save the changes.

   Note: Rmiurl is used for accessing SOA EJBs deployed on SOA managed servers.

   This is the application server URL. (For clustered installation, it is a comma separated list of all the SOA managed server URLs)

   For example:

   t3://mysoa1.oracle.com:8001

   t3s://mysoa1.oracle.com:8001,mysoa2.oracle.com:8002

   t3://mysoa1.oracle.com:8001,mysoa2.oracle.com:8002,mysoa3.com:8003

   Note: Soapurl is used to access SOA web services deployed on SOA managed servers. This is the web server/load balancer URL, in case of a SOA cluster front ended with web server/load balancer. In case of single SOA server, it can be application server URL.

For example,

http://myoimsoa.oracle.com

https://mysoa.oracle.com: 8001

12.3.1.5 Configuring SSL for Design Console

To change the Design console to establish secure connection between Oracle Identity Manager and Design console:

1. Add WebLogic server jars required to support SSL.
2. Copy webserviceclient+ssl.jar from:

   $WL_HOME/server/lib

   to

   $*OIM_HOME*/designconsole/ext directory.
3. Use the Server trust store in the Design console. To access this:
   a. Go to WebLogic Server Administrative console, Environment, Servers.
   b. Click on <OIM_SERVER_NAME> to view details of the Oracle Identity Manger server.
   c. Click the KeyStores tab and note down the "Trust keystore" location in the "Trust" section.

If Design Console is Deployed on the Oracle Identity Manager Host

Set the TRUSTSTORE_LOCATION environment variable to the location of the "Trust keystore" location noted above.

For example:

setenv TRUSTSTORELOCATION/scratch/user1/dogwoodsh100520/beahome/wlserver_10.3/server/lib/DemoTrust.jks

If Design Console is Deployed on a Different Computer than Oracle Identity Manager

Copy the "Trust keystore" to the box in which Design console is present and set the TRUSTSTORE_LOCATION env variable to the location where "Trust keystore" is copied on the local box.

12.3.1.6 Configuring SSL for Oracle Identity Manager Utilities

Oracle Identity Manager client utilities include PurgeCache, GenerateSnapshot, UploadJars, and UploadResources.

Set the TRUSTSTORE_LOCATION environment variable to the location of the "Trust keystore" location.

> **Note:** Refer "Configuring SSL for Design Console" on page 12-13 for details about setting the TRUSTSTORE_LOCATION environment variable to the location of the 'Trust keystore' location.

For example:

setenv
TRUSTSTORELOCATION/scratch/user1/dogwoodsh100520/beahome/wlserver_10.3/server/lib/DemoTrust.jks

12.3.1.7 Configuring SSL for MDS Utilities

All Oracle Identity Manager MDS Utilities which contains WLST scripts must be set to the following environment variable in the shell in which you are running the script:

WLST_PROPERTIES=-Dweblogic.security.SSL.ignoreHostnameVerification=true-Dweblogic.security.TrustKeyStore=DemoTrust

> **Note:** Once this property is set, WLST works fine. You will see INFO/NOTICE messages, which you can ignore.

12.3.1.8 Configuring SSL for SPML/Callback Domain

To configure SSL for SPML/callback domain:

1. Ensure that Oracle Identity Manager port is SSL enabled with HostName verification is set to false.
2. If you are using WebLogic default trust store, you must not change anything other than enabling the SSL mode.
3. If you have certificates other than default, then the trusted certificates should be exchanged between them to establish two-way trust.
4. If you are using a stand-alone client for sending SPML requests for testing purpose, then you must:
   a. Add the following system properties to SPML client command to send the request to SSL enabled OIM port.
      - Djavax.net.ssl.trustStore=D:\Oracle\Middleware1\wlserver_10.3\server\lib\DemoTrust.jks
      - -Djava.protocol.handler.pkgs=weblogic.net
      - -Dweblogic.security.TrustKeyStore=DemoTrust
   b. Add webserviceclient+ssl.jar to your client classpath.

12.3.2 Enabling SSL for Oracle Identity Manager DB

You need to perform the following configurations to enable SSL for Oracle Identity Manager DB:

- Setting Up DB in Server-Authentication SSL Mode
- Creating KeyStores and Certificates
- Updating Oracle Identity Manager
- Updating WebLogic Server

12.3.2.1 Setting Up DB in Server-Authentication SSL Mode

To set up DB in Server-Authentication SSL mode:

1. Stop the DB server and the listener.
2. Configuring the listener.ora file as follows:

a. Navigate to the path:

$DB_ORACLE_HOME/network/admin directory

For example:

/scratch/user1/production-database/product/11.1.0/db_1/network/admin

b. Edit the listener.ora file to include SSL listening port and Server Wallet Location.

The following is the sample listener.ora file:

```
# listener.ora Network Configuration File:
/scratch/rbijja/production-database/product/11.1.0/db_1/network/admin/liste
ner.ora
# Generated by Oracle configuration tools.

SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY =
/scratch/rbijja/production-database/product/11.1.0/db_1/bin/server_keystore
_ssl.p12)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCPS)(HOST = dadvmh0175.us.oracle.com)(PORT =
2484))
    )
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = TCP)(HOST = dadvmh0175.us.oracle.com)(PORT =
1521))
    )
  )

TRACE_LEVEL_LISTENER = SUPPORT
```

3. Configure the sqlnet.ora file as follows:

a. Navigate to the path:

$DB_ORACLE_HOME/network/admin directory

For example:

/scratch/user1/production-database/product/11.1.0/db_1/network/admin

b. Edit sqlnet.ora file to include:

- TCPS Authentication Services
- SSL_VERSION
- Server Wallet Location
- SSL_CLIENT_AUTHENTICATION type (either true or false)
- SSL_CIPHER_SUITES that can be allowed in the communication (optional)

The following is the sample sqlnet.ora file:

```
# sqlnet.ora Network Configuration File:
/scratch/rbijja/production-database/product/11.1.0/db_1/network/admin/sqlne
t.ora
# Generated by Oracle configuration tools.

SQLNET.AUTHENTICATION_SERVICES= (BEQ, TCPS)

SSL_VERSION = 3.0

SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY =
/scratch/rbijja/production-database/product/11.1.0/db_1/bin/server_keystore
_ssl.p12)
    )
  )
```

4. Configure the tnsnames.ora file as follows:

   a. Navigate to the path:

      $DB_ORACLE_HOME/network/admin directory

      For example:

      /scratch/user1/production-database/product/11.1.0/db_1/network/admin

   b. Edit the tnsnames.ora file to include SSL listening port in the description list of the service.

      The following is the sample tnsnames.ora file:

```
# tnsnames.ora Network Configuration File:
/scratch/user1/production-database/product/11.1.0/db_1/network/admin/tnsnam
es.ora
# Generated by Oracle configuration tools.

PRODDB =
 (DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCPS)(HOST = dadvmh0175.us.oracle.com)(PORT =
2484))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = proddb)
    )
  )
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = dadvmh0175.us.oracle.com)(PORT =
1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = proddb)
    )
  )
 )
```

5. Start/Stop utilities for DB server.
6. Start the DB server.

12.3.2.2 Creating KeyStores and Certificates

You can create server side and client side KeyStores using the orapki utility. This utility will be shipped as a part of Oracle DB installation.

KeyStores could be of any format such as JKS and PKCS12. The format of keystore changes based on the provider implementation. For example, JKS is the implementation provided by Sun Oracle where as PKCS12 is implemented by OraclePKIProvider.

Only JKS client KeyStore is used in Oracle Identity Manager for DB server. This is because using non-JKS KeyStores format such as PKCS12 requires significant changes on the installer side at the critical release time. However, Oracle Identity Manager already has a KeyStore named default-KeyStore.jks, which is in JKS format.

The following are the KeyStores that you can create using orapki utility:

- Creating a Root CA Wallet
- Creating DB Server Side Wallet
- Creating Client Side Wallet

> **Note:** Wallets and KeyStores are interchangeably used and they both mean the same. These refer to a repository of public/private keys and self-signed/trusted certificates.

Creating a Root CA Wallet

To create a root certification authority (CA) wallet:

1. Navigate to the following path:

   $DB_ORACLE_HOME/bin directory

2. Create a wallet by using the command:

   ```
   ./orapki wallet create -wallet CA_keystore.p12 -pwd welcome1
   ```

3. Add a self signed certificate to the CA wallet by using the command:

   ```
   ./orapki wallet add -wallet CA_keystore.p12 -dn 'CN=root_test,C=US' -keysize
   2048 -self_signed -validity 3650 -pwd welcome1
   ```

4. View the wallet using the command:

   ```
   ./orapki wallet display -wallet CA_keystore.p12 -pwd welcome1
   ```

5. Export the self signed certificate from the CA wallet using the command:

   ```
   ./orapki wallet export -wallet CA_keystore.p12 -dn 'CN=root_test,C=US' -cert
   self_signed_CA.cert -pwd welcome1
   ```

Creating DB Server Side Wallet

To create a DB server side wallet:

1. Create a server wallet using the command:

   ```
   ./orapki wallet create -wallet server_keystore_ssl.p12 -auto_login -pwd
   ```

```
welcome1
```

2. Add a certificate request to the server wallet using the command:

```
./orapki wallet add -wallet server_keystore_ssl.p12/ -dn
'CN=Customer,OU=Customer,O=Customer,L=City,ST=NY,C=US' -keysize 2048 -pwd
welcome1
```

3. Export the certificate request to a file, which will be used later for getting it signed using the root CA signature:

```
./orapki wallet export -wallet server_keystore_ssl.p12/ -dn
'CN=Customer,OU=Customer,O=Customer,L=City,ST=NY,C=US' -request server_creq.csr
-pwd welcome1
```

4. Get the server wallet's certificate request signed using the CA signature:

```
./orapki cert create -wallet CA_keystore.p12 -request server_creq.csr -cert
server_creq_signed.cert -validity 3650 -pwd welcome1
```

5. View the signed certificate using the command:

```
/orapki cert display -cert server_creq_signed.cert -complete
```

6. Import the trusted certificate in to the server wallet using the command:

```
./orapki wallet add -wallet server_keystore_ssl.p12 -trusted_cert -cert
self_signed_CA.cert -pwd welcome1
```

7. Import this newly created signed certificate (user certificate) to the server wallet using the command:

```
./orapki wallet add -wallet server_keystore_ssl.p12 -user_cert -cert
server_creq_signed.cert -pwd welcome1
```

Creating Client Side Wallet

To create a client side (Oracle Identity Manager server) wallet:

1. Create a client keystore using default-keystore.jks keystore which is populated in the following path:

   DOMAIN_HOME/config/fmwconfig

   Note: You can also use Oracle PKCS12 wallet as the client keystore.

2. Import the self-signed CA trusted certificate that you have already exported using the server side commands, to the client keystore (default-keystore.jks) by using the command:

```
keytool -import -trustcacerts -alias dbtrusted -noprompt -keystore
default-keystore.jks -file self_signed_CA.cert -storepass xellerate
```

12.3.2.3 Updating Oracle Identity Manager

You need to perform the following steps in Oracle Identity Manager to enable Oracle Identity Manager and Oracle Identity Manager DB in SSL mode for a secure communication:

1. Import the trusted certificate into the default-keystore.jks keystore of Oracle Identity Manager.
2. Log in to Enterprise Manager.
3. Navigate to Identity and Access, OIM.
4. Right click and navigate to System MBean Browser.
5. Under Application Defined MBeans, navigate to oracle.iam, Application:oim, XMLConfig, Config, XMLConfig.DirectDBConfig, and DirectDB.
6. Change the values for attributes "Sslenabled", "Url" and click **Apply**. If SSL mode is enabled for DB, then "Url" should contain TCPS enables and SSL port in it.

   For example:

   url="jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCPS)(HOST=my.domain.com)(PORT=2484))(CONNECT_DATA=(SERVICE_NAME=proddb)))"
7. Restart the Oracle Identity Manager server.

12.3.2.4 Updating WebLogic Server

After enabling SSL for Oracle Identity Manager DB, you need to change the following Oracle Identity Manager datasources and authenticators to use DB SSL port:

- Configuring Datasource
- Updating Datasource oimJMSStoreDS Configuration
- Updating Datasource oimOperationsDB Configuration
- Updating Datasource Related to Oracle Identity Manager MDS Configuration
- Updating Oracle Identity Manager Authenticators

Configuring Datasource

To configure the datasource:

1. Log in to Enterprise Manager.
2. Perform the host/port changes.

> **Note:** Before performing changes to database host/port, you must shutdown the managed servers hosting Oracle Identity Manager application. However, you can keep the WebLogic Admin Server up and running.

Updating Datasource oimJMSStoreDS Configuration

To update the datasource oimJMSStoreDS configuration:

1. Log in to WebLogic Server.
2. Navigate to Services, JDBC, Data Sources, oimJMSStoreDS.
3. Click the **Connection Pool** tab and change the value of the URL and Properties to reflect the changes to DB host/port.

Updating Datasource oimOperationsDB Configuration

To update the Change Datasource oimOperationsDB Configuration:

1. Log in to Enterprise Manager.
2. Navigate to Services, JDBC, Data Sources, oimJMSStoreDS.
3. Click the **Connection Pool** tab and change the value of the URL and Properties to reflect the changes to DB host/port.

Updating Datasource Related to Oracle Identity Manager MDS Configuration

To update datasource related to Oracle Identity Manager MDS configuration:

1. Log in to Enterprise Manager.
2. Navigate to Services, JDBC, Data Sources, mds-oim.
3. Click the **Connection Pool** tab and change the value of the URL and Properties to reflect the changes to DB host/port.

Note: You might have to perform similar updates for SOA/OWSM related datasources if required.

Updating Oracle Identity Manager Authenticators

The existing Oracle Identity Manager authenticators in the WebLogic server are configured against Non-SSL DB details and they do not use datasources for communicating with Oracle Identity Manager DB. In order to use SSL DB details in the authenticators, you must perform the following:

1. Ensure that Datasources are configured to SSL.
2. In WebLogic Administrative console, navigate to Security Realms, myrealm, Providers.
3. Remove OIMAuthenticationProvider.
4. Create an authentication provider of type "OIMAuthenticator" and mark the control flag as SUFFICIENT.
5. Create an authentication provider of type "OIMSignatureAuthenticator" and mark the control flag as SUFFICIENT.
6. Reorder the authenticators as:
   a. DefaultAuthenticator
   b. OIMAuthenticator
   c. OIMSignatureAuthenticator
   d. Other providers if any
7. Restart all servers.

12.3.3 Enabling SSL for LDAP Synchronization

You need to perform the following configurations to enable Oracle Identity Manager to use SSL enabled Oracle Virtual Directory (OVD):

- Enabling OVD-OID with SSL
- Updating Oracle Identity Manager for OVD Host/Port

12.3.3.1 Enabling OVD-OID with SSL

To enable OVD-OID with SSL:

1. Log in to the OVD EM console.
2. Expand **Identity and Access** and navigate to ovd1, Administration, Listeners.
3. Click **Create** and enter all the required fields.

> **Note:** You must select the Listener Type as LDAP.

4. Click **OK**.
5. Select the newly created LDAP listener and click **Edit**.
6. In the Edit Listener - OIM SSL ENDPOINT page, edit the newly created LDAP listener.
7. Click **OK**. The SSL Configuration page opens.
8. Select the **Enable SSL** checkbox.
9. In the Advanced SSL Settings section, for SSL Authentication, select **No Authentication**.
10. Click **OK**.
11. Stop and start the OVD server for the changes to take effect.

> **Note:** You must not use the restart option.

12.3.3.2 Updating Oracle Identity Manager for OVD Host/Port

When LDAPSync is enabled on Oracle Identity Manager, Oracle Identity Manager connects with directory servers through OVD. It connects using ldap/ldaps protocol.

To change OVD host/port:

1. Log in to Oracle Identity Manager Administrative and User console.
2. Navigate to Advanced and click **Manage IT Resource**.
3. Select IT Resource Type as **Directory Server** and click **Search**.
4. In the IT Resource Directory Server, edit "server URL" to include SSL protocol and SSL port details.
5. Ensure that Use SSL is set to true and click **Update**.

Part III

Configuration

This part describes the configuration tasks in Oracle Identity Manager.

It contains the following chapters:

13

Configuring User Attributes

The Oracle Identity Manager user management feature is configured and customized by using the configuration management feature. Configuration management helps customize the User Management UI and configure the user entity operations and attributes.

In Oracle Identity Manager, there are certain operations involved in the life-cycle management of each entity. Some of the basic operations for the user entity are:

- Create
- View/Modify
- Browse
- Delete
- Disable
- Enable
- Bulk Operations

> **See Also:** "Managing Users" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about the operations related to the user entity

Based on the operations performed on an entity, a set of attributes is shown to the user in the Oracle Identity Administration. For example, for searching users through advanced search, a set of searchable user attributes is displayed for performing the search. After the search operation is completed, search results involving a set of attributes are displayed. These attribute sets are managed by using the configuration management feature.

You use the Configuration Management UI in the Oracle Identity Administration to define user entity data structure and configure user management operations and attributes. The availability of configuring attributes in the UI is subject to permissions that are controlled by authorization policies. See "User Management" and "Authenticated Self Service" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about authorization policies for managing users and self service operations.

This chapter describes user configuration management in the following sections:

- Synchronizing User-Defined Fields Between Oracle Identity Manager and LDAP
- Configuration Management Architecture

13.1 Entity Configuration Operations

Entity configuration operations allow you to define the set of attributes for the user entity. You can also add the attribute definitions and modify the existing ones. In addition to the attributes defined by default, you can define your own attributes for the user entity.

Note: To access the Configuration Management section in the Advanced Administration, the user must have authorization to configure the user attributes. For more details, see "User Management Configuration" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager*.

Entity configuration operations include:

- Listing Entity Attributes
- Creating Entity Attributes
- Modifying Entity Attributes
- Deleting Entity Attributes
- Performing Category Configuration

13.1.1 Listing Entity Attributes

To list the entity attributes in the Configuration Management console:

1. Login to the Oracle Identity Manager Advanced Administration.
2. In the Welcome page, under Configuration, click **User Configuration**. Alternatively, you can click the Configuration tab, and then click the User Configuration tab.
3. On the left pane of the console, from the Actions menu, select **User Attributes**. The User Attributes page is displayed with a table containing all user attributes that are defined in the User.xml configuration file.

   Table 13 1 describes the columns in the User Attributes table:

Table 13 1 Columns in the User Attributes Table

Column	Description
Category Name	The category to which the attribute belongs. The categorization is used to organize data in the User Management console. **Note:** For information about each category, see "Performing Category Configuration" on page 13-9.
Attribute Names	The unique name for the attribute. It is also used as the caption when this attribute is displayed on the user profile page.
Order in Category	The order of the attributes within the category. The attributes are displayed on the User Management console based on this order.

Table 13 1 (Cont.) Columns in the User Attributes Table

Column	Description
Attribute Type	Whether the type of the attribute is System or user-defined field (UDF). System attributes cannot be deleted and have restrictions on their modifications.
Backend Data Type	The data type of the attribute in the backend datastore.
Display Type	The display type of the attribute in the User Management console.

You can select a row in the User Attributes table, and perform entity configuration operations, such as creating or modifying attributes, which are described in the subsequent sections.

Note: Any administrator user cannot access the Configuration Management section in Oracle Identity Manager Administration. The user must have authorization to configure the user attributes.

4. In the Category Name column, expand a category name by clicking the icon to the left of the category name. The attributes under the category are listed in the Attribute Name column.

13.1.2 Creating Entity Attributes

To create new attributes for an entity:

1. In the User Attributes page, from the Actions menu, select **Create Attribute**. The Create Attribute wizard is displayed.
2. In the Set Attribute Details page of the wizard, enter values in the fields. Table 13 2 lists the fields in the Set Attribute Details page:

Table 13 2 Fields in the Set Attribute Details Page

Field	LOV Types	Description
Attribute Name		This is the unique name for the attribute. It is also used as the caption when this attribute is displayed on the User profile page.
Backend Attribute Name		This is the name of the field that will be created in the user backend schema to store the value specified for this attribute while creating or modifying users . Oracle Identity Manager automatically prefixes the Backend Attribute Name with "USR_UDF.".
Category Name		This is the category name to which the attribute belongs. The categorization is used to organize the data in the UI. **Note:** For information about category configuration, see "Performing Category Configuration" on page 13-9.

Table 13 2 (Cont.) Fields in the Set Attribute Details Page

Field	LOV Types	Description
Display Type		This indicates the display type of the attribute in the UI. This is an attribute property and is stored in the User.xml file as metadata attachment. The available display types are: ■ String ■ Integer ■ Text Area ■ Check Box ■ Double ■ Date ■ Secret ■ List of Values Selecting Display Type sets the appropriate backend and frontend data types. Backend data type is the data type of the attribute in the backend datastore. This is stored in the User.xml file along with the attribute definition. Frontend data type indicates the data type of the attribute as interpreted by Oracle Identity Manager. This is stored in the User.xml file along with the attribute definition. This is not displayed in the UI. See Also: The "Attribute Properties" on page 13-7 section for information about properties to be configured for each attribute
LOV Type		This field is hidden by default. If the display type is selected as List Of Values, then the LOV-related fields are displayed. The LOV Type can be System Generated, Admin Configured, and By Query.
	System Generated	The user can specify existing LOVs. For example: 1. Select **System Generated** as the LOV Type. 2. The LOV Search Options points to the Contains operator by default. In the LOV Code field, enter `country`, and click **Search**. The list of available LOV codes matching the search criteria is displayed in the Avaliable LOV Codes list. 3. Select **Lookup.Locations.Country** and move to the Selected LOV codes list by clicking the right arrow. Only one LOV code should be moved to this list. Then, click **Next**, and complete the rest of the steps in the wizard as described in this section. After saving the attribute, a drop-down list with country codes is displayed in the user details page.

Table 13 2 (Cont.) Fields in the Set Attribute Details Page

Field	LOV Types	Description
	Admin Configured	The user can add this LOV. For example: 1. Select **Admin Configured** as the LOV Type. 2. In the LOV Code field, enter `level`. For a LOV code, you can add multiple LOV options and corresponding LOV descriptions. 3. In the LOV Options field, enter `L1`, and in the LOV Options Description field, enter `Executive`. Then, click **Add**. The LOV option and description is added and are displayed on the page. 4. To add another value, in the LOV Options field, enter `L2`. and in the LOV Options Description field, enter `Senior Executive`. Then click **Add**. 5. After adding multiple values, click **Next**, and complete the rest of the steps in the wizard as described in this section. After saving the attribute, a drop-down list with the values specified in the LOV Options Description field are displayed in the user details page.
	By Query	The LOV Code and LOV Options fields are not displayed. Instead, the following fields are displayed: - LOV Query: In this field, you can specify any SQL query that is valid in the Oracle Identity Manager database schema. - LOV Column to Display: This is a list showing all the columns from the select query. The selected column values are available on clicking a search icon on the pages for creating or modifying the user entity. For example, you might want to display Manager Name instead of Manager Key. - LOV Column to Save: This is a list showing all columns from the select query. The selected column value is the one that is saved in the backend store when the user makes a selection in the dropdown available on the pages for creating or modifying the user entity. For example, you can display Manager Name, but want to save Manager Key value. **Note:** A list of values is already defined in the LKU and LKV tables in the database. For administrator specified, the user must specify an LOV code. This is stored in the LKU table. Associated with each code are the list of values. The user must add new values here. These values are stored in the LKV table and are used as this attribute's LOV values. For system generated, the user can search for LOV codes, and then select a code. Values already exist for this code in the LKV table and are used as this attribute's LOV values. The following is an example of setting the By Query LOV type: 1. Select **By Query** as the LOV Type. 2. In the LOV Query field, enter `SELECT USR_FIRST_NAME as FirstName , USR_LOGIN as UserLogin FROM USR WHERE USR_STATUS = 'Active'`. 3. In the LOV Column to Display list, select **FIRSTNAME**. 4. In the LOV Column to Save list, select **USERLOGIN** and click **Next**, and complete the rest of the steps in the wizard as described in this section. After saving the attribute, a search icon against this attribute is displayed in the user details page. The user can search and select value for the attribute. FIRSTNAME is displayed in the user details page and USERLOGIN is saved in the backend store.

Table 13 2 (Cont.) Fields in the Set Attribute Details Page

Field	LOV Types	Description
LOV Code		This is the code to identify the LOV. For system-generated LOV, this value must be of an existing LOV code. **Note:** The LoV Code, LOV Options, and LOV Options Description fields are displayed only when Display Type is selected as List Of Values. For other display types, these fields are not displayed.
LOV Options		This is displayed only if the LOV Type is administrator specified. The user must specify the LOV values here.
LOV Options Description		These are the descriptive LOV options.

Note: You cannot remove a value from the list of values.

3. Click **Next**. The Set the attribute properties page is displayed.
4. Enter values for the attribute properties. Table 13 3 lists the fields in the Set Properties page:

Table 13 3 Fields in the Set Properties Page

Field	Description
Read Only Value	Determines if the attribute is a read only attribute
Encryption	Determines if the attribute value is stored in encrypted or clear formats
Visible	Determines if the attribute is displayed on the UI
Attribute Size	The maximum size the attribute value can take
Searchable	Determines if the attribute is searchable
Bulk Updatable	Determines if the attribute can be modified while modifying multiple users at the same time.
Default Value	The default value of the attribute to be displayed on the user details.

5. Click **Next**. The Confirm page of the Create Attribute wizard is displayed with information that you entered for creating the attribute.
6. Review the attribute information, and then click **Save**. The MDS schema, which is the User.xml file, and the DB schema are updated with the new attribute. The new attribute added is displayed in the User Management section based on the properties set. See "User Management" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about authorization policies for the user management.

Note:

- To make the newly created attribute that can be viewed or modified in the User Profile, you must create appropriate authorization policies. See "Managing Authorization Policies" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about authorization policies.
- For information about using these fields with LDAP, see "Synchronizing User-Defined Fields Between Oracle Identity Manager and LDAP" on page 13-13.
- For information about configuring request datasets, see "Step 1: Creating a Request Dataset for the Resources" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

13.1.2.1 Attribute Properties

For each attribute, you must configure the following properties:

- **Required:** Determines if every user in the repository must have a non-null value for this attribute. For predefined users, the required attributes have values. If you create a user, you must provide a value for the required attribute. An attribute cannot be modified to required unless the attribute has values for all the existing users.
- **Read-Only:** Makes an attribute read-only, which means that the attribute cannot be modified irrespective of the authorization policy. Some attributes in the UI must always be read-only. These include the system-controlled attributes and may include custom attributes.
- **System Controlled:** Determines if the value can only be set and edited by Oracle Identity Manager.
- **Encrypted:** Determines if the value is stored in the repository in reversible encrypted or clear formats.
- **Searchable:** Determines if the values can be used in simple as well as advanced searches. An attribute must be configured for use in simple search or advanced search by modifying the search configuration. See "Search Operation Configuration" on page 13-10 for information about configuring search operations.
- **Bulk Updatable:** Determines if the attribute can be updated during a bulk modify operation.
- **Size:** Indicates the max size that the value for this attribute can take.
- **Default Value:** The default value of the attribute, which is the value that will be populated in the backend store if no value is provided while creating the user entity.

Note: When you create a new user-defined attribute (UDF), you must add a corresponding entry in any custom resource bundle. The naming convention for the entry is:

global.udf.*BACKEND_UDF_NAME=DESCRIPTION_DISPLAYED_ON_THE_UI*

For example: global.udf.USR_UDF_ATT=Attestation

After adding the entry, upload the resource bundle to MDS by using the Upload JAR utility. See "Upload JAR Utility" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for information about this utility.

13.1.3 Modifying Entity Attributes

The Modify Attribute operation allows you to edit the attributes specific to user entity. To do so:

1. In the User Attributes table, select an attribute.
2. From the Actions menu, select **Modify Attribute**. The Modify Attribute page is displayed.
3. On the Modify Attribute page, edit the attribute details and attribute properties. You cannot edit the Attribute Name and Display Type fields. If you are the system administrator, you can edit all fields except Frontend Attribute Name, Backend Attribute Name, and Backend Data Type.
4. (Optional) Click Preview User Profile to display a preview of the user profile.

   The Preview User Profile feature renders a hypothetical page that contains all available categories and attributes. This feature helps you review the Profile before saving it to the database. Note that a user may not be able to view all of the categories and attributes shown due to user permissions and other constraints.
5. Click **Save** to save the changes.

For attributes with default values, only the following modifications can be done:

- Modifying the default value of the attribute.
- Modifying the visible property of the attribute.
- If an attribute has a default value and is nonrequired, then that attribute can be changed to be required. If an attribute is nonrequired and it does not have a default value, then the attribute cannot be changed to required. Therefore, if you have a nonrequired attribute and you wish to change it to required but it does not have a default value, then add a default value to it first, and then you are able to change the attribute from nonrequired to required.

13.1.4 Deleting Entity Attributes

The Delete operation allows you to delete an attribute. To delete an attribute:

1. In the User Attributes table, select a row.
2. From the Actions menu, select **Delete Attribute**. A message box is displayed asking for confirmation.
3. Click **OK**. A message is displayed confirming that the attribute is deleted.

On performing the delete operation, the actual attribute in the backend is not deleted. The existing data is not affected and audit logs continue to display the data. The deletion happens only in the MDS schema (User.xml).

Note: Default attributes cannot be deleted. Only user-defined attributes can be deleted.

13.1.5 Performing Category Configuration

Category configuration allows you to organize the data in the UI. The following categories are available by default:

- **Basic User Information:** This contains the user's personal information such as first name, last name, e-mail, and organizational information, for example manager or department.
- **Account Settings:** This contains the user login and password information.
- **Account Effective Dates:** The dates on which the user account is activated or deactivated.
- **Provisioning Dates:** The dates on which the user account is provisioned and deprovisioned.
- **Lifecycle:** This is for attributes for user account locked, manually locked, or the date when the account will be automatically deleted. These are not displayed on the UI.
- **System:** These include attributes that are used internally by the application, such as login attempts by the user, the date when the user is created, and user password cannot be changed. These are not displayed on the UI.
- **Other User Attributes:** This contains the remaining attributes of the user.
- **Custom Attributes:** This is an empty category. Attributes are added here by the Deployment Manager while importing from Oracle Identity Manager release 9.1.0 UDFs.
- **Preferences:** This contains the attributes that control the user preferences. For example, Locale and Timezone.

You can perform the following category configuration operations:

- Creating Category
- Renaming Category
- Deleting Category
- Ordering Attributes Within a Category

13.1.5.1 Creating Category

Create category operation allows you to add new categories. To create a new category:

1. In the User Attributes page, from the Actions menu, select **Add Category**. The Create Category dialog box is displayed.
2. In the Category Name field, enter the name of the category.
3. Click **Save** to create the category. A message is displayed stating that the category is successfully created.
4. Click **OK**.

13.1.5.2 Renaming Category

The category names that are displayed in the UI are taken from the resource bundles. To change the display name of a category, you must change the value in the resource bundle.

13.1.5.3 Deleting Category

You can delete only empty categories. To delete a category:

1. In the User Attributes page, select an empty category that you want to delete.
2. From the Actions menu, select **Delete Category**. A message box is displayed asking for confirmation.
3. Click **OK**. A message is displayed that confirms the deletion.
4. Click **OK**.

13.1.5.4 Ordering Attributes Within a Category

You can specify the order of the attributes within the category. The attributes are displayed on the User Management section based on this order.

To order the attributes within a category:

1. In the User Attributes page, select a category whose attributes you want to order.
2. From the Actions menu, select **Order Category Attribtues**. The Order Category Attributes dialog box is displayed with all the attribute names within the selected category.
3. Edit the numbers corresponding to each attribute to specify the attribute's order in the category.
4. Click **Save**.

13.2 Search Operation Configuration

The search operation allows searching of user entities based on a query provided by the user. You can configure the attributes for the search operation, the search results table, and the full table for simple/advanced search.

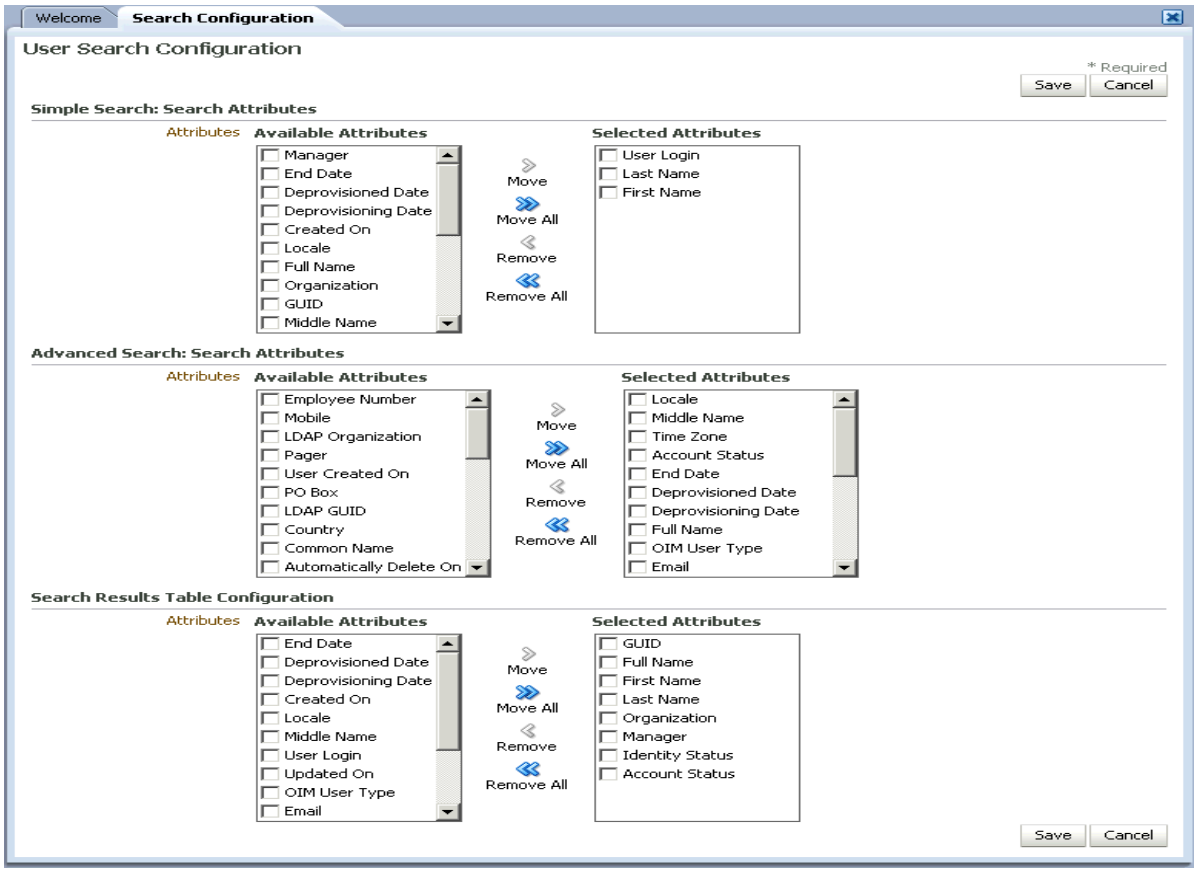
Searchable attributes define the set of attributes to which the search string is applied when performing the simple search. By default, the display name, user name, first name, and last name searchable attributes are configured for simple search. The same are configured by default for advanced search.

Result attributes define the set of attributes that is returned by the search operation. You can define the columns to display in the search results, and the subset to display in the limited search result table for simple search.

You can configure the available attributes for use in simple search and advanced search queries. In addition, you can configure the attributes that you want to be displayed in the search results table. To do so:

1. On the left pane in the User Configuration section, from the Actions menu, select **Search Configuration**. The User Search Configuration page is displayed, as shown in Figure 13 1:

Figure 13 1 The Search Configuration Form



2. In the Simple Search: Search Attributes section, select the attributes that you want to make available for simple search. Click the move and move all icons to add the attributes for simple search. You can also click the remove and remove all icons to remove attributes from the search.
3. In the Advanced Search: Search Attributes section, select the attributes that you want to make available for advanced search. Click the move and move all icons to add the attributes for advanced search.
4. In the Search Results Table Configuration section, select the attributes that you want to display in the search results table. Click the move and move all icons to add the attributes for the search results table.
5. Click **Save**.

Note:

- The Modify and Create operations are not configurable to this level. All the attributes are displayed as editable on the User Management UI, with the following exception:

  ```
  Attributes with property Visible=No
  Attributes with property System Controlled=Yes"
  ```

- The attributes that are visible, but have the property `System Controlled=Yes`, are displayed as read only.
- The final list of attributes displayed on the UI depends on the authorization policies configured.
- Any user defined field is not displayed in the Available Attributes list for simple search.

13.3 User Configuration Management Authorization

Authorization of the user configuration management is governed by a default authorization policy. Custom authorization policies cannot be created for this feature.

> **See Also:** "User Management Configuration" in the *Oracle Fusion Middleware User's Guide for Oracle Identity Manager* for information about the default authorization policy for user configuration management

The users that are members of the System Administrators role are authorized to perform all user configuration operations. The operations are defined by the permissions set for the default authorization policy for this feature. Table 13 4 lists the permissions:

Table 13 4 Authorization Permissions

Permission	Description
Create Attribute	Decides if adding attributes is enabled in the UI for the user. This permission is also used at the API level to decide if the user can add an attribute.
Update Attribute	Decides if updating all attributes is enabled in the UI for the user. This permission is also used at the API level to decide if the user can update attributes.
Delete Attribute	Decides if deleting an attribute is enabled in the UI for the user. This permission is also used at the API level to decide if user can delete an attribute.
Add Category	Decides if adding categories is enabled in the UI for the user. This permission is also used at the API level to decide if the user can add a category.
Order Category Attribute	Decides if updating attributes is enabled in the UI for the user. This permission is also used at the API level to decide if the user can update a category.
Delete Category	Decides if deleting categories is enabled in the UI for the user. This permission is also used at the API level to decide if the user can delete a category.

Table 13 4 (Cont.) Authorization Permissions

Permission	Description
Add Derived Attributes	Decides if adding derived attributes is enabled for the user. The option to add derived attributes is available at the API level only.
Set Search Attributes	Decides if searching configuration is enabled in the UI for the user. This permission is also used at the API level to decide if the user can update simple search and advanced search, and search table attributes.

13.4 Synchronizing User-Defined Fields Between Oracle Identity Manager and LDAP

This section describes how to synchronize user-defined fields between Oracle Identity Manager and LDAP. After creating a user-defined field using the Oracle Identity Manager Advanced Administration Configuration Service, you must extend the OVD and OID schema by adding the new attribute before you can synchronize that attribute. For example, assume you created an Oracle Identity Manager attribute named Employee ID and that the corresponding column name in the USR table is USR_EMPLOYEE_ID. You must add the Employee ID attribute to the orclIDXPerson objectclass in both OVD and OID.

> **See Also:** OVD and OID documentation for information about adding new attributes to the schema.

Use the following steps to synchronize the attribute:

1. Extend the OVD and OID schemas by adding the emplyeeid attribute to the orclIDXPerson objectclass in both OVD and OID.
2. To propagate the attribute value from Oracle Identity Manager to LDAP, perform the following steps:
   a. Export the following file from MDS:

      /metadata/iam-features-ldap-sync/LDAPUser.xml

   b. Add the following entry to the end of the <entity-attributes> tag:

```
<attribute name="Employee ID">
                    <type>string</type>
                    <required>false</required>
                    <attribute-group>Basic</attribute-group>
                    <searchable>true</searchable>
      </attribute>
```

> **Note:** Oracle Identity Manager does not support provisioning or reconciling Boolean-type attributes to LDAP.

   c. Add the following entry to the end of the <target-fields> tag:

```
<field name="employeeid">
              <type>string</type>
              <required>false</required>
       </field>
```

d. Add the following entry to the end of the <attribute-maps> tag:

```
<attribute-map>
              <entity-attribute>Employee ID</entity-attribute>
              <target-field>employeeid</target-field>
       </attribute-map>
```

e. Import the LDAPUser.xml file back into MDS. After importing, verify that the full path in MDS is /metadata/iam-features-ldap-sync/LDAPUser.xml.

3. To propagate the attribute value from LDAP to Oracle Identity Manager, perform these steps:

a. Extend the RA_LDAPUSER table by adding a new column. For example, add the RECON_EMPLOYEE_ID column.

b. Export the reconciliation profile, /db/LDAPUser from MDS.

c. Add the following entry to the end of the <reconFields> tag:

```
<reconAttr>
              <oimFormDescriptiveName>Employee ID</oimFormDescriptiveName>
              <reconFieldName
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xsi:type="xs:string">employeeid</reconFieldName>
              <reconColName>RECON_EMPLOYEE_ID</reconColName>
              <emDataType>string</emDataType>
              <formFieldType/>
                   <targetattr keyfield="false" encrypted="false"
 required="false"
                   type="String" name="usr_employee_id"/>
       </reconAttr>
```

d. Add the following entry to the end of the <reconToOIMMappings> tag:

```
<reconAttr>
              <oimFormDescriptiveName>Employee ID</oimFormDescriptiveName>
              <reconFieldName
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

     xmlns:xs="http://www.w3.org/2001/XMLSchema"
                  xsi:type="xs:string">employeeid</reconFieldName>
              <reconColName> RECON_EMPLOYEE_ID </reconColName>
              <emDataType>string</emDataType>
              <formFieldType/>
              <targetattr keyfield="false" encrypted="false"
required="false"
 type="String" name="

         usr_employee_id">
              <Transformation name="OneToOne">
              <Parameter name=" employeeid " fieldname=" employeeid "/>
              </Transformation>
              </targetattr>
    </reconAttr>
```

e. Import the xml file back into MDS. After importing, verify that the full path in MDS is /db/LDAPUser.

f. Export the /db/RA_LDAPUSER.xml file from MDS.

g. Add the following entry to the end of the <entity-attributes> tag:

```
<attribute name="Employee ID">
              <type>string</type>
              <required>false</required>
              <attribute-group>Basic</attribute-group>
              <searchable>true</searchable>
     </attribute>
```

h. Add this entry to the end of the <target-fields> tag:

```
<field name=" RECON_EMPLOYEE_ID">
              <type>string</type>
              <required>false</required>
     </field>
```

i. Add the following entry to the end of the <attribute-maps> tag:

```
<attribute-map>
              <entity-attribute>Employee ID</entity-attribute>
              <target-field> RECON_EMPLOYEE_ID </target-field>
     </attribute-map>
```

j. Import the RA_LDAPUSER.xml file back into MDS. After importing, verify that the full path in MDS is /db/RA_LDAPUSER.xml.

13.5 Configuration Management Architecture

For all attribute definitions and the Configuration Management pages in the UI, the configuration file for maintaining the user entity attributes is User.xml. This configuration file defines all attributes of user entity and their properties. The mapping of the attribute to the backend attributes or columns is also specified in the file. The attributes to be displayed on the UI are determined based on the attribute properties. For example, if an attribute is system-controlled, then the attribute is not displayed in the UI.

Example 13 1 the code for a sample User.xml configuration file:

Example 13 1 Entity XML Definition

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.oracle.com/schema/oim/entity"
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:tns="http://www.oracle.com/schema/oim/entity">
<element name="entity-definition"
type="tns:entity-definition-type">
</element>

<complexType name="entity-definition-type">
        <all>
        <element name="entity-type" minOccurs="1" maxOccurs="1">
        <complexType>
        <simpleContent>
        <extension base="string">
        <attribute name="child-entity"
        type="boolean">
        </attribute>
        </extension>
        </simpleContent>
        </complexType>
        </element>
        <element name="description" type="string" maxOccurs="1"
```

```
minOccurs="0">
</element>
<element name="provider-instance"
type="tns:provider-instance-type" minOccurs="1"
maxOccurs="1">
</element>
<element name="container-capability"
type="tns:container-definition-type" maxOccurs="1"
minOccurs="1">
</element>
<element name="entity-attributes" maxOccurs="1"
minOccurs="1">
<complexType>
<sequence>
<element name="attribute"
type="tns:attribute-definition-type" maxOccurs="unbounded"
minOccurs="1">
</element>
</sequence>
</complexType>
</element>
<element name="target-fields" maxOccurs="1" minOccurs="1">
<complexType>
<sequence>
<element name="field"
type="tns:field-definition-type" maxOccurs="unbounded"
minOccurs="1">
</element>
</sequence>
</complexType>
</element>
<element name="attribute-maps" maxOccurs="1"
minOccurs="0">
<complexType>
<sequence>
<element name="attribute-map"
type="tns:attribute-map-definition-type"
maxOccurs="unbounded" minOccurs="1">
</element>
</sequence>
</complexType>
</element>
<element name="child-entities" maxOccurs="1"
minOccurs="0">
<complexType>
<sequence>
<element name="entity"
type="tns:attribute-definition-type" maxOccurs="unbounded"
minOccurs="1">
</element>
</sequence>
</complexType>
</element>
<element name="metadata-attachment" maxOccurs="1"
minOccurs="0">
<complexType>
<sequence>
<element name="metadata"
type="tns:metadata-attachment-type" maxOccurs="unbounded"
minOccurs="0">
```

```
        </element>
        </sequence>
        </complexType>
        </element>
        <element name="control-attributes" minOccurs="0" maxOccurs="1">
        <complexType>
        <sequence>
        <element name="attribute" minOccurs="1" maxOccurs="unbounded">
        <complexType>
        <sequence>
        <element name="type" type="string"
        minOccurs="1" maxOccurs="1">
        </element>
        <element name="description"
        type="string" minOccurs="0" maxOccurs="1">
        </element>
        <element name="required"
        type="boolean" minOccurs="1" maxOccurs="1">
        </element>
        </sequence>
        <attribute name="name"
        type="string" use="required">
        </attribute>
        </complexType></element>
        </sequence>
        </complexType></element>
        </all>
</complexType>

<complexType name="provider-instance-type">
        <all>
<element name="repository-instance" type="string" maxOccurs="1"
minOccurs="0"></element>
<element name="provider-type" type="string" maxOccurs="1" minOccurs="1"></element>
<element name="parameters" minOccurs="0" maxOccurs="1">
<complexType>
<sequence>
<element name="parameter" maxOccurs="unbounded" minOccurs="1">
<complexType>
<sequence>
<element name="value" type="string" maxOccurs="unbounded" minOccurs="1">
</element>
</sequence>
<attribute name="name" type="string">
</attribute>
</complexType>
</element>
</sequence>
</complexType>
</element>
</all>
</complexType>

<complexType name="parameter-definition-type">
        <all>
<element name="type" type="string" maxOccurs="1" minOccurs="1">
</element>
<element name="description" type="string" maxOccurs="1" minOccurs="0">
</element>
<element name="required" type="boolean" maxOccurs="1" minOccurs="1">
```

```
</element>
<element name="multi-valued" type="boolean" maxOccurs="1" minOccurs="0">
</element>
</all>
<attribute name="name" type="string"></attribute>
</complexType>

<complexType name="attribute-definition-type">
        <all>
        <element name="type" type="string" maxOccurs="1"
        minOccurs="1">
        </element>
        <element name="description" type="string" maxOccurs="1"
        minOccurs="0">
        </element>
        <element name="required" type="boolean" maxOccurs="1"
        minOccurs="1">
        </element>
        <element name="searchable" type="boolean" maxOccurs="1"
        minOccurs="1">
        </element>
        <element name="MLS" type="boolean" minOccurs="0" maxOccurs="1"></element>
        <element name="default-value" type="string" maxOccurs="1"
        minOccurs="0">
        </element>
        <element name="attribute-group" type="string" maxOccurs="1"
        minOccurs="1">
        </element>
        <element name="metadata-attachment" maxOccurs="1"
        minOccurs="0">
        <complexType>
        <sequence>
        <element name="metadata"
        type="tns:metadata-attachment-type" maxOccurs="unbounded"
        minOccurs="0">
        </element>
        </sequence>
        </complexType>
        </element>
        </all>
        <attribute name="name" type="string"></attribute>
</complexType>

<complexType name="field-definition-type">
        <all>
<element name="type" type="string" maxOccurs="1" minOccurs="1">
</element>
<element name="description" type="string" maxOccurs="1" minOccurs="0">
</element>
<element name="required" type="boolean" maxOccurs="1" minOccurs="1">
</element>
</all>
<attribute name="name" type="string"></attribute>
</complexType>

<complexType name="attribute-map-definition-type">
        <all>
<element name="entity-attribute" type="string" maxOccurs="1" minOccurs="1">
</element>
<element name="target-field" type="string" maxOccurs="1" minOccurs="1">
```

```
</element>
</all>
</complexType>

<element name="repository-definition"
type="tns:repository-definition-type">
</element>

<complexType name="repository-definition-type">
        <all>
<element name="name" type="string" maxOccurs="1" minOccurs="1">
</element>
<element name="class" type="string" maxOccurs="1" minOccurs="1">
</element>
<element name="parameters" maxOccurs="1" minOccurs="0">
<complexType>
<sequence>
<element name="parameter-def" type="tns:parameter-definition-type"
maxOccurs="unbounded" minOccurs="1">
</element>
</sequence>
</complexType>
</element>
<element name="description" type="string" maxOccurs="1" minOccurs="0"></element>
</all>
</complexType>

<element name="provider-definition"
type="tns:provider-definition-type">
</element>

<complexType name="provider-definition-type">
        <all>
<element name="name" type="string" maxOccurs="1" minOccurs="1"></element>
<element name="type" maxOccurs="1" minOccurs="1">
<complexType>
<choice>
<element name="DataProvider" type="string"></element>
<element name="RelationProvider" type="string">
</element>
</choice>
</complexType>
</element>
<element name="class" type="string" maxOccurs="1" minOccurs="1"></element>
<element name="description" type="string" maxOccurs="1" minOccurs="0"></element>
<element name="parameters" maxOccurs="1" minOccurs="0">
<complexType>
                    <sequence>
<element name="parameter-def" type="tns:parameter-definition-type"
maxOccurs="unbounded" minOccurs="1">
</element>
</sequence>
</complexType>
</element>
</all>
</complexType>

<element name="repository-instance">
<complexType>
            <all>
```

```
<element name="name" type="string"></element>
<element name="type" type="string"></element>
<element name="parameters" maxOccurs="1" minOccurs="0">
<complexType>
<sequence>
<element name="parameter" maxOccurs="unbounded" minOccurs="1">
<complexType>
<sequence>
<element name="value" type="string" maxOccurs="1" minOccurs="1">
</element>
</sequence>
<attribute name="name" type="string">
</attribute>
</complexType>
</element>
</sequence>
</complexType>
</element>
</all>
</complexType>
</element>

<complexType name="container-definition-type">
        <sequence>
<element name="enabled" type="boolean" maxOccurs="1" minOccurs="1"></element>
<element name="contained-entity" type="string" maxOccurs="unbounded"
minOccurs="0">
</element>
</sequence>
</complexType>

<complexType name="relation-definition-type">
        <all>
<element name="relation-type" type="string" maxOccurs="1" minOccurs="1"></element>
<element name="description" type="string" maxOccurs="1" minOccurs="0"></element>
<element name="provider-instance" type="tns:provider-instance-type" maxOccurs="1"
minOccurs="1">
</element>
<element name="entity1" type="tns:relation-entity-type" maxOccurs="1"
minOccurs="1">
</element>
<element name="entity2" type="tns:relation-entity-type" maxOccurs="1"
minOccurs="1"></element>
<element name="relation-attributes" maxOccurs="1" minOccurs="1">
<complexType>
<sequence>
<element name="attribute" type="tns:attribute-definition-type"
maxOccurs="unbounded" minOccurs="0">
</element>
</sequence>
</complexType>
</element>
<element name="target-fields" maxOccurs="1" minOccurs="1">
<complexType>
<sequence>
<element name="field" type="tns:field-definition-type" maxOccurs="unbounded"
minOccurs="0">
</element>
</sequence>
</complexType>
```

```
</element>
<element name="attribute-maps" maxOccurs="1" minOccurs="0">
<complexType>
<sequence>
<element name="attribute-map" type="tns:attribute-map-definition-type"
maxOccurs="unbounded" minOccurs="1">
</element>
</sequence>
</complexType>
</element>
</all>
</complexType>

<element name="relation-definition"
type="tns:relation-definition-type">
</element>

<complexType name="relation-entity-type">
        <all>
<element name="entity-type" type="string"></element>
<element name="attribute" type="string"></element>
<element name="attribute-in-entity" type="string"></element>
<element name="attribute-group" type="string" maxOccurs="1"
minOccurs="1"></element>
</all>
</complexType>


<element name="datatype-definition"
type="tns:datatype-definition-type">
</element>

<complexType name="datatype-definition-type">
        <all>
<element name="name" type="string" maxOccurs="1" minOccurs="1"></element>
<element name="class" type="string" maxOccurs="1" minOccurs="1"></element>
<element name="base-type" type="string" maxOccurs="1" minOccurs="1"></element>
</all>
</complexType>

<complexType name="metadata-attachment-type">
        <all>
<element name="name" type="string"></element>
<element name="value" type="string"></element>
<element name="category" type="string"></element>
</all>
</complexType>

<element name="derived-datatype-definition"
type="tns:derived-datatype-definition-type">
</element>

<complexType name="derived-datatype-definition-type">
        <all>
<element name="name" type="string" maxOccurs="1" minOccurs="1">
</element>
<element name="class" type="string" maxOccurs="1" minOccurs="1">
</element>
<element name="parameters" minOccurs="0" maxOccurs="1">
<complexType>
```

```
<sequence>
<element name="parameter" maxOccurs="unbounded" minOccurs="1">
<complexType>
<sequence>
<element name="value" type="string" maxOccurs="1" minOccurs="1">
</element>
</sequence>
<attribute name="name" type="string">
</attribute>
</complexType>
</element>
</sequence>
</complexType>
</element>
</all>
</complexType>
</schema>
```

The entity XML files are stored in MDS. When a new attribute is added, the database schema is updated along with the entity XML in MDS. The configuration service APIs can be used to fetch the attribute information and can be leveraged while building custom UI.

14

Managing Password Policies

The Administration folder of Oracle Identity Manager Design Console enables you to administer Oracle Identity Manager.

> **See Also:** *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for detailed information about Oracle Identity Manager Design Console and all the forms available in Oracle Identity Manager Design Console

You can perform the following tasks by using the Administration folder of Oracle Identity Manager Design Console:

- Creating a Password Policy
- Setting the Criteria for a Password Policy

14.1 Creating a Password Policy

You can use the Password Policies form in Oracle Identity Manager Design Console to create password policies, and thereby:

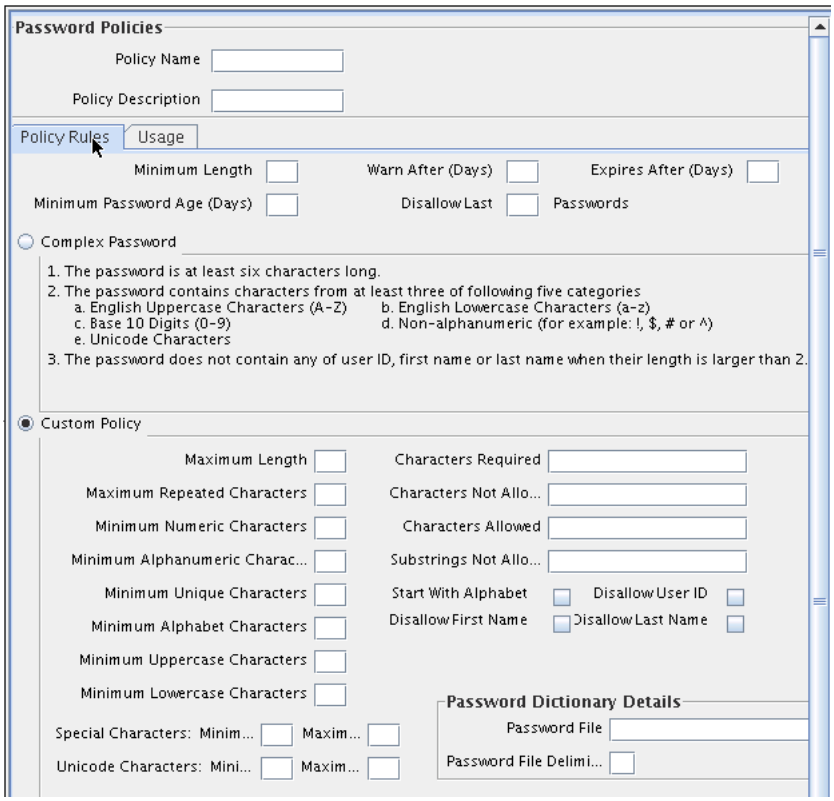
- Set password restrictions, for example, define the minimum and maximum length of passwords
- See rules and resource objects that are associated with a password policy

> **See Also:** "Password Policies Form" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for information about the Password Policies form and the tabs in this form

To create a password policy:

1. Open the Password Policies form. Figure 14–1 shows the Password Policies form.

Figure 14–1 The Password Policies Form



2. In the Policy Name field, enter the name of the password policy.
3. In the Policy Description field, enter a short description of the password policy.
4. Click **Save**.

Note:

- A password policy is not applied during the creation of an Oracle Identity Manager user through trusted reconciliation.
- After you create a password policy, it must be supplied with criteria and associated with a resource. To supply your password policy with criteria, use the Policy Rules tab of this form. To associate your password policy with a resource, use the Password Policies Rule tab of the Resource Object form to create a password policy and rule combination that will be evaluated when accounts are created or updated on the resource. The password policy will be applied when the criteria for the rule are met. Each password policy can be used by multiple resources.

The tabs in this form become functional after you create a password policy. These tabs are used to set the criteria for the password policy and to view the rules and resource objects that are associated with the current password policy. The following sections discuss these tabs:

- The Policy Rules Tab
- The Usage Tab

14.1.1 The Policy Rules Tab

You use the Policy Rules tab to specify criteria for your password policy, for example, the minimum and maximum length of passwords.

You can use either or both of the following methods to set password restrictions:

- Enter information in the appropriate fields, or select the required check boxes. For example, to indicate that a password must have a minimum length of four characters, enter **4** in the **Minimum Length** field.
- In the **Password File** field, enter the directory path and name of the password policy file (for example, `c:\xellerate\userlimits.txt`). This file contains predefined words that you do not want to be used as passwords. The delimiter specified in the **Password File Delimiter** field separates these words. the predefined words in the file cannot be used as passwords. For example, if the file contains the word welcome, then welcome, Welcome, and welcome123 are invalid passwords

Figure 14–1 shows the **Policy Rules** tab of the Password Policies form.

Table 14–1 describes the data fields on the **Policy Rules** tab. You specify the password policy criteria in these fields.

Note: If a data field of the policy is empty, a password conforming to this policy does not have to meet the criteria of that field for the password to be valid. For example, when the **Minimum Numeric Characters** data field is blank, Oracle Identity Manager will accept a password, regardless of the number of characters included in it.

Table 14–1 Fields of the Policy Rules Tab of the Password Policies Form

Field Name	Description
Minimum Length	The minimum number of characters that a password must contain for the password to be valid. For example, if you enter **4** in the **Minimum Length** field, the password must contain at least four characters. This field accepts values from 0 to 999.
Expires After Days	The duration in days for which users can use a password. For example, if you enter **30** in the **Expires After Days** field, users must change their passwords by the thirtieth day from when it was created or last modified. **Note:** After the number of days specified in the **Expires After Days** field passes, a message is displayed asking the user to change the password. This field accepts values from 0 to 999.
Disallow Last Passwords	The frequency at which old passwords can be reused. This policy ensures that users do not change back and forth among a set of common passwords. For example, if you enter **10** in the **Disallow Last Passwords** field, users are allowed to reuse a password only after using 10 unique passwords. This field accepts values from 0 to 24.

Table 14–1 (Cont.) Fields of the Policy Rules Tab of the Password Policies Form

Field Name	Description
Warn After (Days)	The number of days that must pass before a user is notified that the user's password will expire on a designated date. For example, suppose you enter **30** in the **Expires After Days** field, and **20** in the **Warn After (Days)** field, and the password is created on November 1. On November 21, the user will be informed that the password will expire on December 1. This field accepts values from 0 to 999.

On the Policy Rules tab of the Password Policies form, you can configure either a complex password or custom password policy. If you select the **Complex Password** option, you cannot use the Custom Password option setup and passwords will be evaluated against the complex password criteria that you enter on the Policy Rules tab.

The remaining fields in the Policy Rules tab are discussed in the following sections:

- Complex Password
- Custom Policy

Complex Password

The following are the complex password criteria:

- The password is at least six characters long. This password length overrides the **Minimum Length** field if the value entered in the **Minimum Length** field is less than 6. For example, if you enter 2 in the **Minimum Length** field, at least six characters will be required for the password because it must have at least six characters according to the complex password criteria.
- The password must contain characters from at least three of the following five categories:
  - English uppercase characters (A - Z)
  - English lowercase characters (a - z)
  - Base 10 digits (0 - 9)
  - Non-alphanumeric characters (for example: !, $, #, or %)
  - Unicode characters
- The password must not contain the user's first name, last name, or user ID when their length is greater than 2.

  The names are parsed for delimiters: commas, periods, dashes or hyphens, underscores, spaces, pound signs, and tabs. If any of these delimiters are found, then the names are split and all sections are verified not to be included in the password. For example, if the user name is john-d, then d will not be checked in the password because its length is less than 2. Similarly, if the name is John Richard Doe, then the password cannot contain john, richard, or doe.

  When checking against the user's full name, characters such as commas, periods, dashes or hyphens, underscores, spaces, pound signs, and tabs are treated as delimiters that separate the name into individual character sets. Each character set that has three or more characters is searched in the password. If the character set is present in the password, the password change is rejected. For example, the name `John Richard-Doe` is split into three character sets: `John`, `Richard`, and `Doe`.

This user cannot have a password that consists of three continuous characters from either `John` or `Richard` or `Doe` anywhere in the password. However, the password can contain the substring `d-D` because the hyphen (-) is treated as the delimiter between the substrings `Richard` and `Doe`. In addition, the search for character sets in the password is not case-sensitive.

Note: If the user's full name is less than three characters in length, the password is not checked against it because the rate at which passwords will be rejected is too high.

Custom Policy

If you select the **Custom Policy** option, you can set a custom password policy by using the fields listed in Table 14–2.

Table 14–2 Fields of the Policy Rules Tab for Setting Custom Password Policy

Field Name	Description
Maximum Length	The maximum number of characters that a password can contain. For example, if you enter **8** in the **Maximum Length** field, a password is not accepted if it has more than eight characters. This field accepts values from 1 to 999.
Maximum Repeated Characters	The maximum number of times a character can be repeated in a password. For example, if you enter **2** in the **Maximum Repeated Characters** field, a password is not accepted if any character is repeated more than two times. For example, `RL112211` would not be a valid password because the character `1` is repeated three times. **Note:** In this example, there are four occurrences of the character `1`, which means that it is repeated three times. This field accepts values from 1 to 999.
Minimum Numeric Characters	The minimum number of digits that a password must contain. For example, if you enter **1** in the **Minimum Numeric Characters** field, a password must contain at least one digit. This field accepts values from 0 to 999.
Minimum Alphanumeric Characters	The minimum number of letters or digits that a password must contain. For example, if you enter **6** in the **Minimum Alphanumeric Characters** field, a password must contain at least six letters or numbers. This field accepts values from 0 to 999.
Minimum Unique Characters	The minimum number of nonrepeating characters that a password must contain. For example, if you enter **1** in the **Minimum Unique Characters** field, a password is accepted if at least one character in the password is not repeated. For example, `1a23321` would be a valid password because the character `a` in the password is not repeated although the remaining characters are repeated. This field accepts values from 0 to 999.

Table 14–2 (Cont.) Fields of the Policy Rules Tab for Setting Custom Password Policy

Field Name	Description
Minimum Alphabet Characters	The minimum number of letters that a password must contain. For example, if you enter **2** in the **Minimum Alphabet Characters** field, the password is not accepted if it has less than two letters. This field accepts values from 0 to 999.
Special Characters: Minimum	The minimum number of non-alphanumeric characters (for example, #, %, or &) that a password must contain. For example, if you enter **1** in the **Special Characters: Minimum** field, a password must have at least one non-alphanumeric character. This field accepts values from 0 to 999.
Special Characters: Maximum	The maximum number of non-alphanumeric characters that a password can contain. For example, if you enter **3** in the **Special Characters: Maximum** field, a password is not accepted if it contains more than three non-alphanumeric characters. This field accepts values from 1 to 999.
Minimum Uppercase Characters	The minimum number of uppercase letters that a password must contain. For example, if you enter **8** in the **Uppercase Characters: Minimum** field, a password is not accepted if it contains less than eight uppercase letters. This field accepts values from 0 to 999.
Minimum Lowercase Characters	The minimum number of lowercase letters that a password must contain. For example, if you enter **8** in the **Minimum Lowercase Characters** field, a password is not accepted if it has less than eight lowercase letters. This field accepts values from 0 to 999.
Unicode Characters: Minimum	The minimum number of Unicode characters that a password must contain. For example, if you enter **3** in the **Unicode Characters: Minimum** field, the password is not accepted if it has less than three Unicode characters. This field accepts values from 0 to 999.
Unicode Characters: Maximum	The maximum number of Unicode characters that a password can contain. For example, if you enter **8** in the **Unicode Characters: Maximum** field, a password is not accepted if it has more than eight Unicode characters. This field accepts values from 1 to 999.

Table 14–2 (Cont.) Fields of the Policy Rules Tab for Setting Custom Password Policy

Field Name	Description
Characters Required	The characters that a password must contain. For example, if you enter **x** in the **Characters Required** field, a password is accepted only if it contains the character x. The character you specify in the **Characters Required** field, must be mentioned in the **Characters Allowed** field. In addition, if you specify more than one character, then do not provide delimiters. Commas and white spaces are also considered as characters in this field. For example, if you specify characters such as a,x,c, then the password is not accepted unless it contains comma.
Characters Not Allowed	The characters that a password must not contain. For example, if you enter an exclamation point (!) in the **Characters Not Allowed** field, a password is not accepted if it contains an exclamation point.
Characters Allowed	The characters that a password can contain. For example, if you enter the percent sign (*%)* in the **Characters Allowed** field, a password is accepted if it contains a percent sign, given that all other criteria are met. **Note:** If any character is used in the password and that character is not in the Characters Allowed field, then the password will be rejected. For example, if the Characters Allowed field has "abc" and the password is "dad", then the password is rejected because "d" is not in the Characters Allowed field. If you specify the same character in the **Characters Allowed** and **Characters Not Allowed** fields, an error message is returned when you create the password policy.
Substrings Not Allowed	A series of consecutive alphanumeric characters that a password must not contain. For example, if you enter **IBM** in the **Substrings Not Allowed** field, a password is not accepted if it contains the letters I, B, and M, in successive order.
Start With Alphabet	Whether or not the password begins with a letter. For example, if you select this option, then the password 123welcome is not accepted because the password does not begin with a letter.
Disallow User ID	This check box specifies if the user ID will be accepted as the whole password or as part of the password. When this check box is selected, a password will not be valid if the user ID is entered in the **Password** field. In addition, the password is not valid if the user ID occurs as a part of the password specified in the Password field. If you deselect this check box, the password will be accepted, even if it contains the user ID.
Disallow First Name	This check box specifies if the user's first name will be accepted as the whole password or as part of the password. When this check box is selected, a password will not be valid if the user's first name is entered in the **Password** field. In addition, the password is not valid is the first name is entered as a part of the password. If you deselect this check box, the password will be accepted, even if it contains the user's first name.

Table 14–2 (Cont.) Fields of the Policy Rules Tab for Setting Custom Password Policy

Field Name	Description
Disallow Last Name	This check box specifies if the user's last name will be accepted as the whole password or as part of the password. When this check box is selected, a password will not be valid if the user's last name is entered in the **Password** field. In addition, the password is not valid is the last name is entered as a part of the password. If you deselect this check box, the password is accepted, even if it contains the user's last name.
Password File	The path and name of a file that contains predefined terms, which are not allowed as passwords. **Note:** If settings on the **Policy Rules** tab differ from the specifications in the password file, Oracle Identity Manager will use the settings on the **Policy Rules** tab.
Password File Delimiter	The delimiter character used to separate terms in the password file. For example, if a comma (,) is entered in the Password File Delimiter field, the terms in the password file will be separated by commas.

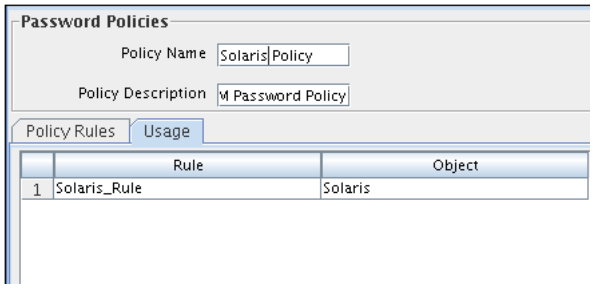
You can attach a process form with one of the Password fields to a resource. A password entered for a resource is validated against the password policy associated with that resource.

14.1.2 The Usage Tab

You use this tab to view the rules and resource objects that are associated with the current password policy.

Figure 14–2 shows the **Usage** tab of the Password Policies form. In this example rules are being defined for the **Solaris** password policy.

Figure 14–2 Usage Tab of the Password Policies Form



See Also: "Password Policies Rule Tab" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for more information about the relationship between password policies and resource objects

14.2 Setting the Criteria for a Password Policy

You can attach a process form with one of the Password fields to a resource. If you apply a password policy to the same resource and create an access policy for the resource, the password entered by the user in the process form is not validated against

the password policy rules. This is because when a resource is provisioned to the user, the user must provide the password, which will be validated against the password policy rules applied to the resource.

To set the criteria for a password policy:

1. Open the required password policy definition.
2. Click the **Policy Rules** tab.
3. Either enter information into the appropriate fields, or select the required check boxes.
4. Click **Save**.

15

Managing Identity and Resource Information

This chapter describes managing users in Oracle Identity Manager Design Console. It contains the following sections:

15.1 Overview of User Management

The User Management folder provides tools to create and manage information about a company's organizations, users, roles, and resources.

This folder contains the following forms:

- **Organizational Defaults**: Use this form to view records that reflect the internal structure of your organization and to designate information related to these entities.
- **Policy History**: Use this form to view user records that your employees require.
- **Roles**: Use this form to view records for roles, called user groups in earlier releases of Oracle Identity Manager, to whom you can assign some common functionality.

15.2 Managing Organization Information

The Organizational Defaults form is in the User Management folder. You use this form to view records that reflect the structure of your organization and to enter and modify information related to organizational entities. An organization record contains information about an organizational unit, for example, a company, department, or branch.

A suborganization is an organization that is a member of another organization, for example, a department in a company. The organization that the suborganization belongs to is referred to as a parent organization.

You use the Organizational Defaults tab to specify default values for parameters on the custom process form for resources that can be provisioned for the current organization. Each process form is associated with a resource object that is allowed for the organization, or with a resource that has the Allow All option on the associated Resource Objects form selected.

The values that you provide on the Organizational Defaults tab become the default values for all users in the organization. Oracle recommends that you do not specify default values for passwords and encrypted parameters.

Figure 15–1 shows the Organizational Defaults form.

Figure 15–1 Organizational Default Form

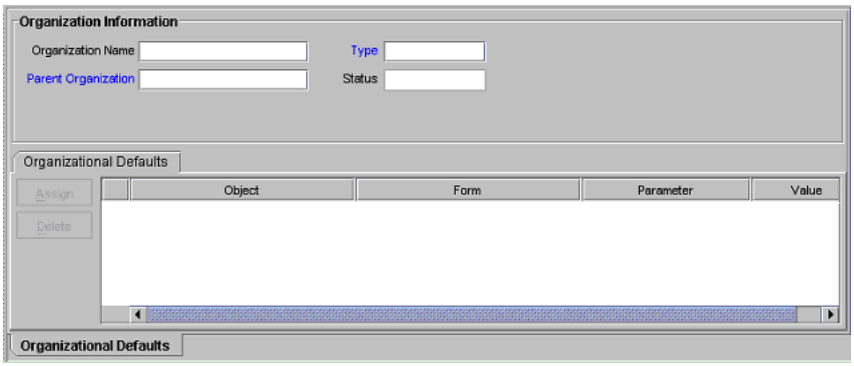


Table 15–1 describes the fields of the Organizational Default form.

Table 15–1 Fields of the Organizational Defaults Form

Field Name	Description
Organization Name	Name of the organization.
Type	The classification type of the organization, for example, Company, Department, Branch.
Status	The current status of the organization (Active, Disabled, or Deleted).
Parent Organization	The organization to which this organization belongs. If a parent organization is displayed in this field, this organization is displayed on the Sub Organizations tab for the parent organization. If this field is empty, this organization is a top-level organization.

15.3 Viewing Resources Allowed or Disallowed for Users

You use the Policy History form to view information about the resources that are allowed or disallowed for a user.

There are two types of users in Oracle Identity Manager:

- **End-user administrators**: This user can access Oracle Identity Manager Design Console and the Oracle Identity Manager Administrative and User Console. The system administrator sets permissions to enable end-user administrators to access a subset of the forms in Oracle Identity Manager Design Console.
- **End-users**: This user can access only the Oracle Identity Manager Administrative and User Console and generally has fewer permissions than end-user administrators. Only resource objects that are defined as self-service on the Objects Allowed tab of the user's organization are available for provisioning requests by using the Oracle Identity Manager Administrative and User Console.

Table 15–2 shows this form.

Figure 15–2 Policy History Form

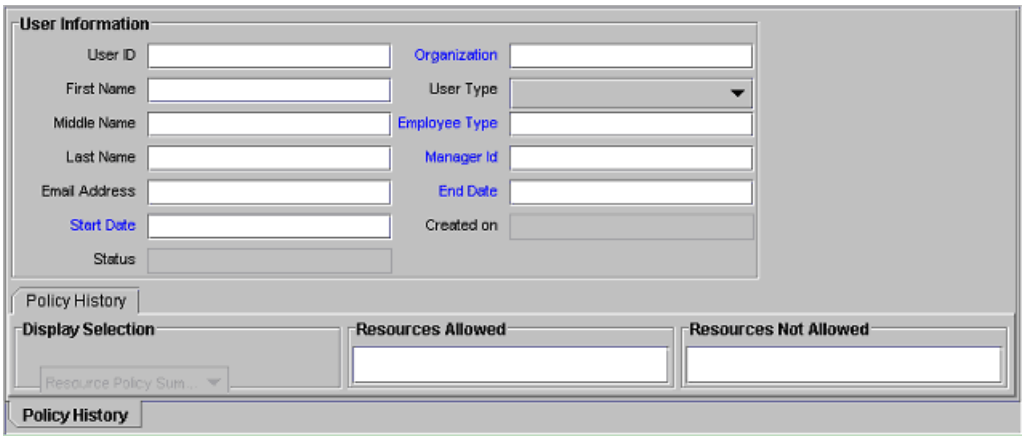


Table 15–2 describes the fields of the Policy History form.

Table 15–2 Fields of the Policy History Form

Field Name	Description
User ID	The user's Oracle Identity Manager login ID.
First Name	The user's first name.
Middle Name	The user's middle name.
Last Name	The user's last name.
Email Address	The user's e-mail address.
Start Date	The date on which the user's account will be activated.
Status	The current status of the user (Active, Disabled, or Deleted).
Organization	The organization to which the user belongs.
User Type	The user's classification status. Valid options are End-User and End-User Administrator. Only end-user administrators have access to Oracle Identity Manager Design Console.
Employee Type	The employment status of the user at the parent organization (for example, full-time, part-time, intern, and so on).
Manager ID	The user's manager.
End Date	The date on which the user's account will be deactivated.
Created on	The date and time when the user record was created.

15.3.1 Policy History Tab

Use this tab to view resource objects that are allowed or disallowed for a user, based on the following:

- Access policies for the user group to which the user belongs
- Resource objects that are allowed by the organization to which the user belongs

The Policy History tab contains a Display Selection region. To organize the contents of this tab, go to the uppermost box in this region and select an item from one of its menus, as follows:

- **Resource Policy Summary**: Displays resource objects that are allowed or disallowed based on the user's organization and applicable access policies.

- **Not Allowed by Org**: Displays only resource objects that are disallowed, based on the user's organization.
- **Resources by Policy**: Displays a second box that contains the access policies for the user groups to which the user is a member.

  Select an access policy from this box to display the resource objects that are allowed or disallowed for the user, based on this access policy.

A tracking system enables you to view resources that are allowed or disallowed for a user, based on the organizations the user is a member of and the access policies that apply to the user.

The resource objects that are allowed for the user are displayed in the Resources Allowed list. This list represents resource objects that can be provisioned for the user. It does not represent the resource objects that are provisioned for the user.

The resource objects that are disallowed for the user are displayed in the Resources Not Allowed list.

To view the tracking system:

1. Go to the Policy History tab.
2. Find the Display Selection region on this tab.
3. Click **Policy History**.

From the User Policy Profile History window, you can view resources that are allowed or disallowed for a user for the date and time you selected, as follows:

- From the **History Date** box, you can select a date.
- From the **Display Type** box, you can display resources that are allowed or disallowed based on the organizations the user is a member of, the access policies that apply to the user, or both.
- From the **Policy** box, you can display the access policy that determines what resource objects are allowed or disallowed for the user.

15.4 Assigning Role Entitlements

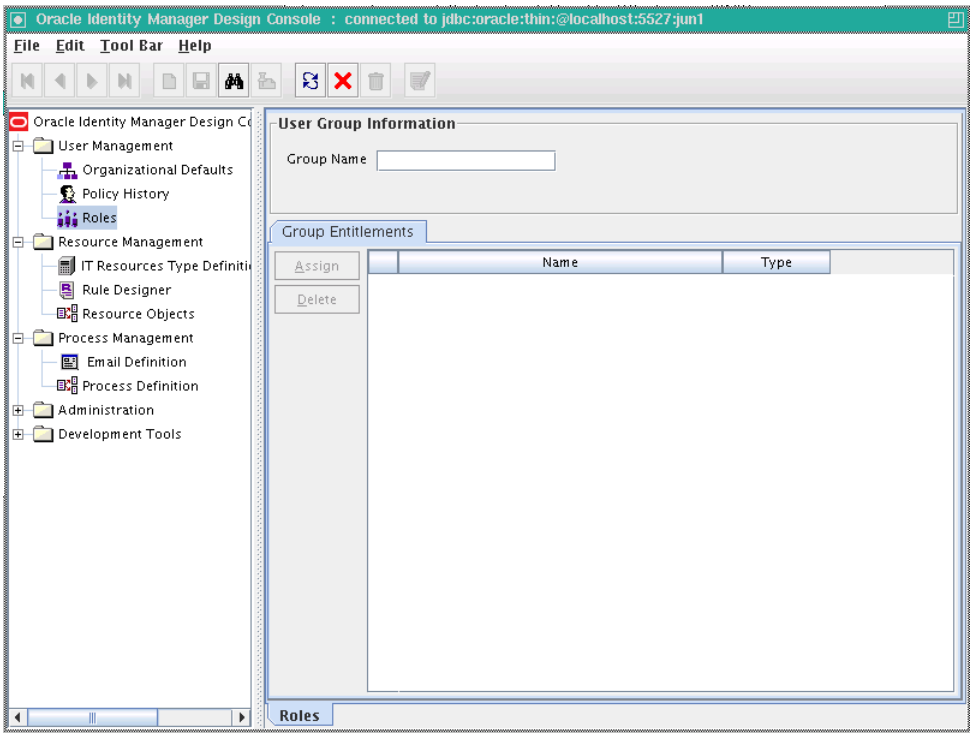
The Group Entitlements form is displayed in the User Management folder. You use it to creating and move forms, and to designate the forms and folders that members of a role can access through the Explorer.

To designate forms and folders to roles by using the Group Entitlements form:

1. In the Explorer, double-click **Group Entitlements**.

   The User Group Information page is displayed, as shown in Figure 15–3:

Figure 15–3 Roles Form



2. In the **Group Name** field, enter the name of the role.
3. Click **Assign**.

   The User Form Assignment lookup table is displayed.
4. From the lookup table, select the user form for this role.

   Use the arrow buttons to either add or delete from the **Assigned Forms** list.
5. Click **OK**.

   The newly added user forms are listed in a Group Entitlements table. The Group Entitlements Table displays all available roles. This table shows the name of the user form and the type. In the Group Entitlements table, there are two types, javaform and folder. A **javaform** is a Java-based, graphical interface. A **folder** is a container of one or many javaforms.

> **See Also:** "Default Roles" for information about pre-existing roles in Oracle Identity Manager

16

Managing Asynchronous Execution

This chapter describes the AsyncService provided by the Oracle Identity and Access Management (IAM) platform and contains the following topics:

- Section 16.1, "Overview of AsyncService"
- Section 16.2, "Async Routing and Configuration"
- Section 16.3, "Troubleshooting Failed Async Tasks"
- Section 16.4, "Working with the Diagnostic Dashboard UI"

16.1 Overview of AsyncService

The AsyncService is one of the services provided by the IAM platform to run tasks asynchronously. Tasks are executed asynchronously to improve performance and throughput.

Some Identity Management operations take a long time to complete. So, it makes sense to split these operations into two parts, a short synchronous interaction followed by a long asynchronous process. The user is provided a response at the end of the synchronous interaction, and the remaining operation is performed asynchronously.

The AsyncService allows the Oracle Identity Manager component to submit tasks for asynchronous execution. The caller then performs other tasks. It is the responsibility of the AsyncService to execute this task whenever the computing resources are available.

16.2 Async Routing and Configuration

The AsyncService uses a configuration file, *async-messaging.xml*, to route and configure Async tasks. This configuration file is stored in the MetaData Store (MDS) schema in Oracle Identity Manager database. The MDS path of the file is /file/async-messaging.xml.

Example 16–1 shows a snippet of the configuration file.

Example 16–1 Sample Configuration File

```
<tns:async-config>
<task-config>
<class>oracle.iam.reconciliation.impl.ActionTask</class>
<destination>queue/oimReconQueue</destination>
</task-config>
<task-config>
<class>com.thortech.xl.schedule.jms.messagetype.AttestationTaskMessage</class>
<destination>queue/oimAttestationQueue</destination>
<priority>NORMAL</priority>
```

```
<maxRetries>2</maxRetries>
</task-config>
<task-config>
<class>com.thortech.xl.schedule.jms.messagetype.AttestationRequestMessage</class>
<destination>queue/oimAttestationQueue</destination>
<priority>HIGH</priority>
</task-config>
<default-config>
<destination>queue/oimDefaultQueue</destination>
<maxRetries>3</maxRetries>
</default-config>
</tns:async-config>
```

To modify the configuration file, import it by using the MDS import utility, make changes in the file, and then export the modified file by using the MDS export utility. For more information about the MDS utilities, see "MDS Utilities" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.

16.2.1 Configuration Parameters

The System Administrators can configure the following parameters in the configuration file for Async tasks:

- **Destination:** You can assign high-volume tasks to their own dedicated queues. For instance, in Example 16–1, all the Async tasks are assigned to the same destination queue `attestationQueue`. You can decide where to send each massage by creating separate destination queues for each Async task.

  Note: You must ensure that the queue exists in the Application Server before assigning a task to it.

- **Priority:** You can set a priority when multiple types of Async tasks are assigned to the same destination queue. Its value can be one of the following:
  - NORMAL
  - HIGH
  - LOW
- **Max Retries:** Async task execution error recovery is handled in two ways, automated and manual. The automated retry mechanism uses a scheduled task to retry all failed tasks at specific intervals. Max Retries parameter allows the System Administrator to specify the maximum number of times a task can be retried in the event of an execution failure. See Section 16.3 for detailed information about error handling and recovery mechanisms.

16.3 Troubleshooting Failed Async Tasks

Errors may occur during execution of tasks or messages. The Async task execution error recovery is a combination of automated retries and manual intervention. If a task encounters an error during task execution, then it is added to a `FailedTasks` table and the System Administrator is notified. See Section 16.3.1 and Section 16.3.2 for detailed information about error handling mechanisms.

16.3.1 Automated Retry Error Handling Mechanism

A scheduled task is provided to automate retries of failed tasks at periodic intervals. The maximum number of times a task is retried by the scheduled task can be configured by using the max-retries property of the async task, as shown in Example 16–2.

Example 16–2 Configuring Max Retries

```
<async-task>
    <class>oracle.iam.reconciliation.impl.ActionTask</class>
    <destination>reconQueue</destination>
    <max-retries>2</max-retries>
</async-task>
```

16.3.2 Manual Retry Error Handling Mechanism

The System Administrator can use the Oracle Identity Manager Diagnostic Dashboard User Interface (UI) to view the failed tasks and retry a task after taking the appropriate remedial action. See Section 16.4 for more information on Oracle Identity Manager Diagnostic Dashboard UI.

16.4 Working with the Diagnostic Dashboard UI

The Diagnostic Dashboard provides a UI for the System Administrator to view and retry failed Async tasks. This section contains the following topics:

- Starting the Diagnostic Dashboard UI
- Viewing Failed Async Tasks
- Retrying Failed Async Tasks
- Resubmitting Failed Async Tasks
- Purging Failed Async Tasks

16.4.1 Starting the Diagnostic Dashboard UI

To start the Diagnostic Dashboard UI:

1. Access the Diagnostic Dashboard home page by using the following URL:

   `http://host:port/XIMDD`

2. Click the **Manage Failed Tasks** link on the left menu pane.
3. Enter the user name and password. The Manage Failed Tasks page is displayed.

> **Note:** You need System Administrator privileges to access the Diagnostic Dashboard UI.

16.4.2 Viewing Failed Async Tasks

The System Administrator can view the details of each failed task, for instance the cause for the task to fail and the remedial action to be undertaken.

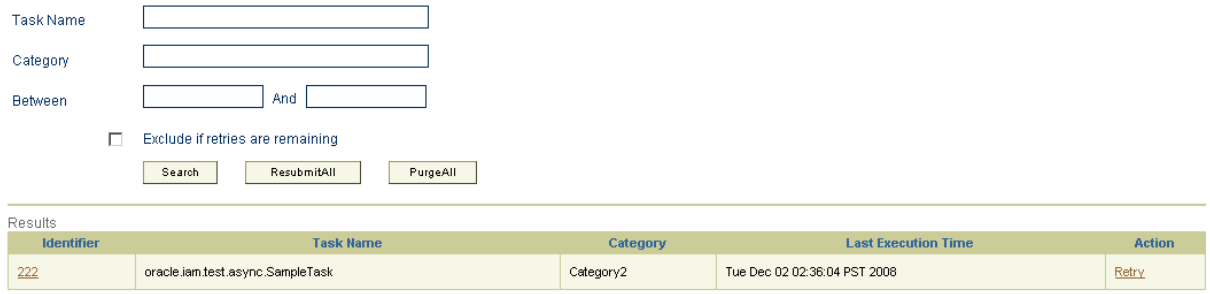
The user can view the details of the failed tasks by either providing the filter criteria or by clicking the **Search** button.

16.4.2.1 To view failed async tasks

1. Log in to the Diagnostic Dashboard main page. See Section 16.4.1 for more information.
2. Perform one of the following to view a list of failed tasks.
   - Click **Search** to view a list of all the failed tasks.
   - Search for the failed task based on the following filter criteria.
     - **Task Name:** Type the name of the failed task.
     - **Category:** Type the category of the failed task.
     - **Between:** Specify the date range.
     - Select the **Exclude if retries are remaining** option if you do not want to view the tasks for which automated retries are still pending.

     Click **Search** after providing the filter criteria. The list of failed async tasks are displayed, as shown in Figure 16–1:

Figure 16–1 Failed Async Tasks



3. Click the Identifier link to view detailed information about the failed task. In this scenario, click `222`. The following information is displayed:
   - Task Name
   - Instance ID
   - Category
   - Last Execution Time
   - Cause
   - Action
   - Stack Trace

16.4.3 Retrying Failed Async Tasks

The System Administrator can retry a specific failed task directly from the Diagnostic Dashboard UI and then view the results of the retry.

16.4.3.1 To retry failed Async task

1. Search for the failed task that you want to retry. See Section 16.4.2.1 for more information.

2. Click the **Retry** link. The retry status for the task is displayed. The following details are provided.
   - Retry Status
   - Task Summary
   - Stack Trace
   - Cause
   - Resolution

16.4.4 Resubmitting Failed Async Tasks

All the failed tasks are resubmitted to the Async queue. These are later executed asynchronously.

To resubmit failed tasks, click **ResubmitAll**.

16.4.5 Purging Failed Async Tasks

There are situations when there are numerous failed Async tasks. The System Administrator might feel that there is no use retrying these tasks. In such a scenario, the failed tasks can be purged. The action purge removes all the failed Async tasks from the database. In other words, there no more tasks to retry.

16.4.5.1 To purge failed Async tasks

1. Search for the failed task that you want to retry. See Section 16.4.2.1 for more information.
2. Click **PurgeAll**.

17

Enabling Offline Provisioning

In online provisioning, multiple provisioning operations are performed in sequence. For example, if you create a request to allocate (provision) five resources to five OIM User, then the system:

- Treats the provisioning of one resource to one user as a provisioning operation
- Processes provisioning operations in sequence, one after the other

Provisioning is treated as a single transaction. This approach could cause performance issues under certain conditions. In addition, there is a higher probability of transaction timeout and, therefore, the entire transaction being rolled back.

In offline provisioning, provisioning operations are converted into JMS messages. One JMS message is submitted for each resource provisioned to each user. For example, if you create a request to provision five resources to five OIM Users, then 25 JMS messages are generated. Processing of each JMS message is treated as a single transaction, and it is asynchronous and independent of other JMS messages. Processing of the other messages continues even if one transaction times out. This approach offers better performance and a lower probability of transaction timeout.

This section discusses the following topics:

17.1 Features of Offline Processing

The following are features of offline provisioning:

- The offline provisioning approach is applied only during Provision (Create Target System Account) Resource, Enable Resource, Disable Resource, and Revoke Resource operations. The offline provisioning approach is not applied in a provisioning operation that involves modification of an allocated (provisioned) resource.
- Offline provisioning is not applied during organization provisioning.
- You enable offline provisioning at the resource object level. The procedure is described later in this chapter.
- JMS messages generated during offline provisioning are processed in parallel. Processing of each JMS message is treated as a single transaction, and it is asynchronous and independent of other JMS messages. This approach provides

better performance over the online provisioning approach in which provisioning operations are processed in sequence.

- The response to a provisioning operation is displayed almost immediately after the provisioning data is submitted. This response is not dependent on the processing of each operation.

  When you view the resource details for a resource instance of an OIM User, you can view the "Provisioning in Queue", "Enable in Queue", "Disable in Queue" and "Revoke in Queue" statuses for Provision, Enable, Disable, and Revoke operations respectively if provisioning for a particular resource has not yet been processed.

- The final status of the resource instance is the same as the status for online provisioning. For example, if a message for a resource is processed successfully, then the Provisioned status is displayed. The same status is displayed for online provisioning.

- Within offline provisioning, processing of each message is treated as an independent transaction. Rejection or failure of a single message does not affect processing of the remaining messages in provisioning.

- During offline provisioning, details of a failed message (along with an explanation) are not displayed on the console. This behavior is different from that of online provisioning in which details of a failed operation are displayed on the console. In offline provisioning, details of failed messages are stored in the Off-line Persistent Store (OPS) table. You can view these details by running the Off-line Resource Provisioning Messages report. See "Reports Related to Offline Provisioning" for information about this report.

- When you disable or delete an OIM User, all the resources provisioned to the user must be disabled or revoked, respectively. This is the expected outcome in both online and offline provisioning. The outcome is the same if provisioning succeeds, regardless of the type of provisioning. However, the outcome is different if an exception is encountered during the operation.

  Online provisioning treats a Disable or Delete OIM User operation as one transaction. If even a single resource cannot be successfully disabled or revoked on the target system, then the entire transaction is rolled back.

  Note: A rollback in Oracle Identity Manager does not affect the status of the resource on the target systems. For example, suppose an OIM User is assigned Resource A, Resource B, and Resource C. If this OIM User is deleted, then the system first tries to delete the resources from the respective target systems. Suppose Resources A and B are deleted but problems are encountered on attempting to delete Resource C. In this case, the entire transaction is rolled back and the status of Resources A, B, and C on Oracle Identity Manager is set to whatever it was at the start of the transaction. However, the actual status of Resources A and B on their target systems is that they have been deleted.

  In offline provisioning, the following JMS messages are generated in response to a Disable or Delete OIM User operation:

  - JMS message to disable or delete the OIM User
  - JMS messages to disable or revoke each resource assigned to the OIM User

If the OIM User is successfully disabled or deleted, then a message (statement) to this effect is displayed on the console. The display of this message (statement) is independent of the success or failure of the JMS messages generated to disable or revoke each resource. If the JMS message for a particular resource fails, then that resource becomes a rogue account in Oracle Identity Manager. You can identify these rogue accounts by running the Off-line Resource Provisioning Messages report. For each of the remaining resources, the status of the resource (Disabled or Revoked) in Oracle Identity Manager is the same as the status of the resource (Disabled or Deleted) on the target system.

17.2 Enabling and Disabling Offline Provisioning

As mentioned earlier, you enable offline provisioning at the resource object level. Off-line provisioning is applicable only when the Auto Save Form option is already selected in the Process Definition form.

To enable offline provisioning:

1. Log in to Oracle Identity Manager Design Console.
2. Expand **Resource Management**, and double-click **Resource Objects**.
3. Search for and open the resource object for which you want to enable offline provisioning.
4. On the Resource Object form, select **Off-line Provisioning**. This enables off-line provisioning for enable, disable, and revoke resource operations.

   When the Off-line Provisioning option is not selected, the specific resource provisioning, enable, disable, and revoke operations occur online.
5. Click the Save icon.

To disable offline provisioning:

1. Log in to Oracle Identity Manager Design Console.
2. Expand **Resource Management**, and double-click **Resource Objects**.
3. Search for and open the resource object for which you want to enable offline provisioning.
4. On the Resource Object form, deselect the **Off-line Provisioning** check box.
5. Click the Save icon.

17.3 Reports Related to Offline Provisioning

When an online provision, enable, disable, or revoke operation fails, the error messages and other information about the operation are displayed on the UI. When an off-line operation fails, the information about the failure are updated in the OPS table. The Offline Resource Provisioning Messages report in Oracle BI Publisher stores all the error messages.

17.4 Configuring the Remove Failed Off-line Messages Scheduled Task

Configure the Remove Failed Off-line Messages scheduled task to schedule deletion of failed provisioning operations from the OPS table. While configuring this scheduled task, set a value for the Remove Failed Messages Older Than (days) attribute.

See Chapter 2, "Managing Scheduled Tasks" for information about working with scheduled tasks.

18

Using Enterprise Manager for Managing Oracle Identity Manager Configuration

Oracle Identity Manager stores the configuration files in MDS. Most of the configurations are exposed as MBeans. Therefore, you can control the configuration values by using Enterprise Manager. In some instances, might have to export the complete files to file system, make the necessary changes, and then import the files back into the repository, as described in the following sections:

18.1 Using MBeans for Configuration Changes

To change configuration settings by using Mbeans:

1. When the administrative server and at least one Oracle Identity Manager managed server is running, login to Enterprise Manager by using the URL in the following format:

   http://*ADMINSTRATION_SERVER*/em

2. Navigate to Identity and **Access**, **oim**. Right-click and navigate to **System MBean Browser**.

3. Under Application Defined MBeans, navigate to **oracle.iam**, **Application:oim**, **XMLConfig**, **Config**.

   All the configuration files are in this location.

18.2 Exporting and Importing Configuration Files

To export or import configuration files:

> **See Also:** "MDS Utilities and User Modifiable Metadata Files" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for information about the list of configuration files that can be exported and imported

1. When the administrative server and at least one Oracle Identity Manager managed server is running, login to Enterprise Manager by using the URL in the following format:

   http://*ADMINSTRATION_SERVER*/em

2. Navigate to Identity and **Access**, **oim**. Right-click and navigate to **System MBean Browser**.
3. Under Application Defined MBeans, navigate to **oracle.mds.lcm**, **Server:oim_server1**, **Application:oim**, **MDSAppRuntime**.
4. To export the configuration files:
   a. Click the **Operations** tab, and then click **exportMetaData**.
   b. In the toLocation field, enter `/tmp` or the name of another directory.
   c. Select createSubDir as **false**.
   d. In the docs field, enter the complete file location as the Element.
   e. Also select **false** for excludeAllCust, excludeBaseDocs, and excludeExtendedMetadata. Then, click **Invoke**.

      This exports the file specified in the docs field to the directory specified in the toLocation field.
5. To import the configuration files:
   a. Click **importMetaData**.
   b. In the fromLocation field, enter `/tmp` or the name of the directory in which you have the configuration files.
   c. Select createSubDir as **false**.
   d. In the docs field, enter the complete file location as the Element. For example, /db/oim-config.xml.
   e. Also select **false** for excludeAllCust, excludeBaseDocs, and excludeExtendedMetadata. Then, click **Invoke**.

      This imports the file specified in the docs field to MDS in the toLocation field.

Part IV

Administrative Utilities

This part describes a number of additional features for Oracle Identity Manager administrators.

It contains the following chapters:

19

Working with the Diagnostic Dashboard

This chapter describes the Diagnostic Dashboard utility shipped with Oracle Identity Manager and contains the following topics:

19.1 Overview of the Diagnostic Dashboard

Diagnostic Dashboard is a stand-alone application that helps you validate some of the Oracle Identity Manager prerequisites and installation.

You must have the appropriate system administrator permissions for your Application Server and Oracle Identity Manager environments to use this tool. You need DBA-level permissions to execute some database-related tests.

Note: The Diagnostic Dashboard and Oracle Identity Manager must be installed on the same application server.

19.2 Installing the Diagnostic Dashboard

The Diagnostic Dashboard utility is distributed on the installation CD-ROM with the Oracle Identity Manager Installer. It is available as a EAR file in the `Diagnostic Dashboard` directory on the CD-ROM.

19.2.1 Installing the Diagnostic Dashboard on Oracle WebLogic Server

This section discusses the steps you need to perform to install the Diagnostic Dashboard on Oracle WebLogic Server.

To install the Diagnostic Dashboard on Oracle WebLogic Server:

1. Log in to Oracle WebLogic Administration Console.
2. In the left navigation pane, click **Deployments**. It lists all the applications deployed on the server.
3. Click **Install**.

4. Navigate to the location for deploying the EAR file. Typically, the EAR file is located in the following directory:

   OIM_ORACLE_HOME/server/webapp/optional/

5. Select **XIMDD.ear** from the **Current Location** panel.
6. Click **Next** on the **Choose targeting style** page.
7. Select **ManagedServer** (Oracle Identity Manager Server) from the **Available targets for XIMDD** panel, and click **Next**.
8. Click **Finish**. The following message appears:

   ```
   All changes have been activated. No restarts are necessary.
   The deployment has been successfully installed.
   ```

   You can access the Diagnostic Dashboard from the following location:

   ```
   http://OIM_server_host_ip:port/XIMDD
   ```

19.3 Starting the Diagnostic Dashboard

After the Diagnostic Dashboard is deployed, you can access it by using a URL of the following format:

http://*OIM_HOST:OIM_PORT*/XIMDD

Log into Diagnostic Dashboard with administrator privileges. Click the **Diagnostic Dashboard** link on the left menu pane to display the Diagnostic Dashboard main page.

The Diagnostic Dashboard utility indicates on which application server the tool is deployed.

19.4 Using the Diagnostic Dashboard

The Diagnostic Dashboard main page includes the sections listed in the following table:

Section	Description
Application Server	Displays the name of the application server
Oracle Identity Manager Installation	Displays installation details such as product version, build number, host, and location of the product
Test Details	Displays the test name and its description
Test Parameters	Displays the parameters required for testing

To run a test:

1. Select the test by selecting the option on the Diagnostic Dashboard main page.
2. Enter the required parameters.
3. Click **Verify** to see the result.

   The Diagnostic Dashboard Test Result page is displayed with the status information listed in the following table.

Test Result	Description
Result Summary	Shows all the selected tests with icons (pass or fail) indicating the result. The test name is a Web link that allows the user to jump to the result details directly.
Test Name	Displays the name of the test
Description	Displays the description of the test
Input Parameters	Displays the parameters of the test
Result	Displays the outcome of the test
Details	Displays details about the outcome of the test

4. Click **Diagnostic Dashboard** on the left menu pane or **Return to Diagnostic Dashboard** to return to the previous test page.

19.5 Running Tests By Using the Diagnostic Dashboard

The following tests are available for different application servers.

- Oracle Database Prerequisites Check
- Database Connectivity Check
- Account Lock Status
- Data Encryption Key Verification
- Scheduler Service Status
- Remote Manager Status
- JMS Messaging Verification
- Target System SSL Trust Verification
- Java VM System Properties Report
- Oracle Identity Manager Libraries and Extensions Version Report
- Oracle Identity Manager Libraries and Extensions Manifest Report
- Test Basic Connectivity
- Test Provisioning
- Test Reconciliation
- SOA-Oracle Identity Manager Configuration Check
- Request Diagnostic Information
- Orchestration Status
- Retry Failed Orchestration
- SPML Web Service
- Test OWSM Setup
- Test SPML to Oracle Identity Manager Request Invocation
- SPML Attributes to Oracle Identity Manager Attributes
- Username Test
- Diagnose Creation of User and Role in Oracle Identity Manager and LDAP

- Diagnose OVD Connection
- Diagnose LDAP Reserve Container

19.5.1 Oracle Database Prerequisites Check

Prerequisite: The following are the prerequisites for running this test:

Prerequisite	Description
Database Server	Enter the location of the database server.
Port	Enter the port number.
Database Name	Enter the database name (SID).
Oracle Identity Manager Database User Name	Enter the Oracle Identity Manager database user name.
System User Name	Enter the system user name.
System User Password	Enter system user password.

Description: Checks if the specified Oracle Database instance meets the prerequisites for Oracle Identity Manager installation. This test requires SYSTEM permissions.

Result: It displays the following information:

- Necessary permissions for user
- XA support enabled
- JVM enabled
- Oracle version Information

19.5.2 Database Connectivity Check

Prerequisite: None

Description: Run this test to verify whether or not Oracle Identity Manager is able to connect to the database. This test verifies the direct database connection and the J2EE data sources (XA).

Result: It displays the following information:

- Direct database connectivity
- XA execution

19.5.3 Account Lock Status

Prerequisite: The following is the prerequisite for verifying this test:

Prerequisite	Description
User Login	Enter the user name.

Description: Oracle Identity Manager locks an account when there are successive multiple invalid login attempts. This test checks whether or not a specified account is locked.

Result: Checks for locked or unlocked accounts in the database.

19.5.4 Data Encryption Key Verification

Prerequisite: None

Description: The data encryption key in an Oracle Identity Manager installation should be the same as the one used to encrypt the data in the Oracle Identity Manager database. This may not be the case when an Oracle Identity Manager installation is pointed to a database schema created for a different Oracle Identity Manager installation. This can also happen when a database dump from one Oracle Identity Manager installation is imported for a different Oracle Identity Manager installation without copying the corresponding key.

Result: Checks if the database key is present in the Oracle Identity Manager configuration directory.

19.5.5 Scheduler Service Status

Prerequisite: None

Description: Checks the status of the Oracle Identity Manager Scheduler Service running on the server.

Result: Displays the status of the scheduler service.

19.5.6 Remote Manager Status

Prerequisite: None

Description: Reports the status of the Remote Managers that this Oracle Identity Manager installation is set to work with.

Result: Displays the status of the Remote Manager.

19.5.7 JMS Messaging Verification

Prerequisite: None

Description: The purpose of this test is to verify that Oracle Identity Manager will be able to submit a JMS message and process it.

Result: Displays if Oracle Identity Manager is able to submit and process a JMS message.

19.5.8 Target System SSL Trust Verification

Prerequisite: The following are the prerequisites for running this test:

Prerequisite	Description
Target System	Enter the host name.
Port	Enter the port number.
Certificate Store Location	Enter the location for storage.
Certificate Store Password	Enter the password for storage.

Description: Oracle Identity Manager must be set up to trust the target system certificates if the connectivity is over Secure Sockets Layer (SSL). Enter the host name and the port where a target system is listening for SSL connections.

Result: It displays the following information:

- Valid and invalid host and port address
- Trusted certificates

19.5.9 Java VM System Properties Report

Prerequisite: None

Description: Displays all the Java VM system properties.

Result: Displays all the Java VM system properties.

19.5.10 Oracle Identity Manager Libraries and Extensions Version Report

Prerequisite: None

Description: Reports all the versions of the Oracle Identity Manager libraries and extensions.

Result: Displays the versions of the Oracle Identity Manager libraries and extensions.

19.5.11 Oracle Identity Manager Libraries and Extensions Manifest Report

Prerequisite: None

Description: Reports the manifest information of the Oracle Identity Manager libraries and extensions.

Result: Displays the manifest information of the Oracle Identity Manager libraries and extensions.

19.5.12 Test Basic Connectivity

Prerequisite: The following are the prerequisites for running this test:

Prerequisite	Description
IT Resource Type Name	Enter the IT resource type.
IT Resource Instance Name	Enter the IT resource name.

Description: Tests the connection to the target system by using the IT resource for the target system.

> **Note:** If the specified IT resource type was created when you deployed a predefined connector, then see the Oracle Identity Manager Connector Pack release notes and documentation to determine whether or not this test can be applied for the release of the connector that you deployed.

Result: Displays the results of the connectivity test. If the test fails, then the cause of the error is also displayed.

19.5.13 Test Provisioning

Prerequisite: The following are the prerequisites for running this test:

Prerequisite	Description
IT Resource Type Name	Enter the IT resource type.
IT Resource Instance Name	Enter the IT resource name.

Description: Performs a basic Create User operation on the target system.

> **Note:** If the specified IT resource type was created when you deployed a predefined connector, then see the Oracle Identity Manager Connector Pack release notes and documentation to determine whether or not this test can be applied for the release of the connector that you deployed.

Result: Displays the results of the provisioning test. Test data created on the target system during the test is deleted at the end of the test.

19.5.14 Test Reconciliation

Prerequisite: The following are the prerequisites for running this test:

Prerequisite	Description
IT Resource Type Name	Enter the IT resource type.
IT Resource Instance Name	Enter the IT resource name.

Description: Performs a basic reconciliation operation on the target system.

> **Note:** If the specified IT resource type was created when you deployed a predefined connector, then see the Oracle Identity Manager Connector Pack release notes and documentation to determine whether or not this test can be applied for the release of the connector that you deployed.

Result: Displays the results of the reconciliation test. Test data reconciled into Oracle Identity Manager during the test is deleted at the end of the test.

19.5.15 SOA-Oracle Identity Manager Configuration Check

Prerequisite: None

Description: Checks whether the details provided for SOA-wiring are valid or not.

Result: Displays the status for the following tests:

1. Validation for SOA connection with Oracle Identity Manager and authentication of user in SOA
2. Authentication and search of Oracle Identity Manager DB user

19.5.16 Request Diagnostic Information

Prerequisite: The following is the prerequisite for running this test:

Prerequisite	Description
Request ID	Enter the ID of the request for which diagnostic information is required

Description: Provides the orchestration ID and the composite details for the given request ID.

Result: Displays the following information:

1. Orchestration process ID associated with the given request ID.
2. Composite details of the request along with details of approval and process task.

19.5.17 Orchestration Status

Prerequisite: The following are the prerequisites for running this test:

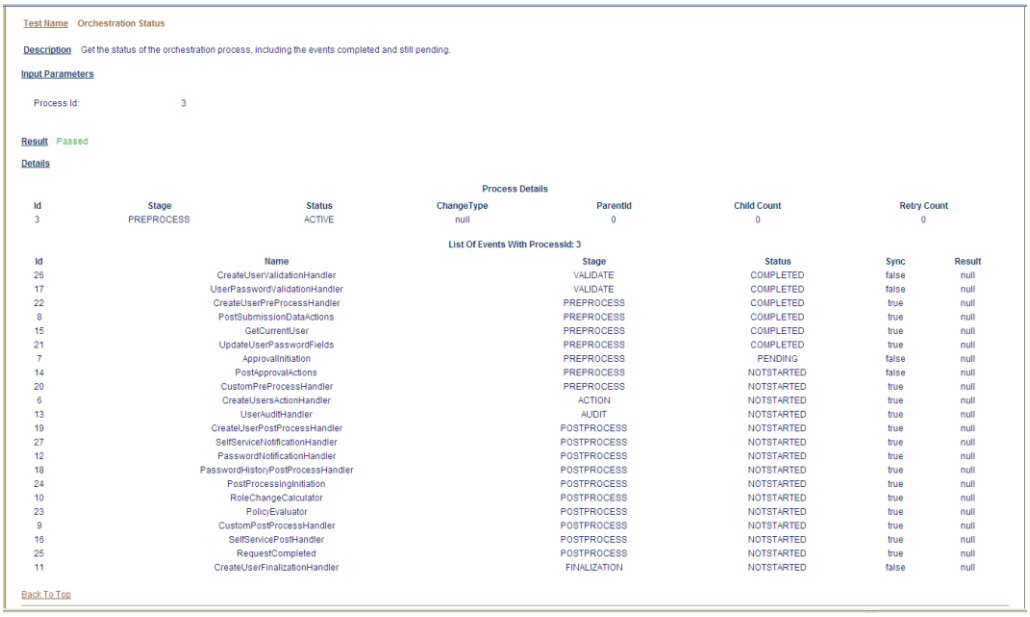
Prerequisite	Description
Process Id	Enter the Id of the orchestration process.

Description: Provides the status of the orchestration process in the Oracle Identity Manager Kernel. It also provides details and status about all the event handlers involved in that process.

Result: Displays the status of the orchestration process as Failed, Completed, or Active.

Figure 19–1 displays the status of the orchestration process, including the events completed and still pending.

Figure 19–1 Sample Output for Orchestration Status Test



Test Name Orchestration Status

Description Get the status of the orchestration process, including the events completed and still pending.

Input Parameters

Process Id: 3

Result Passed

Details

Process Details

Id	Stage	Status	ChangeType	ParentId	Child Count	Retry Count
3	PREPROCESS	ACTIVE	null	0	0	0

List Of Events With ProcessId: 3

Id	Name	Stage	Status	Sync	Result
26	CreateUserValidationHandler	VALIDATE	COMPLETED	false	null
17	UserPasswordValidationHandler	VALIDATE	COMPLETED	false	null
22	CreateUserPreProcessHandler	PREPROCESS	COMPLETED	true	null
8	PostSubmissionDataActions	PREPROCESS	COMPLETED	true	null
15	GetCurrentUser	PREPROCESS	COMPLETED	true	null
21	UpdateUserPasswordFields	PREPROCESS	COMPLETED	true	null
7	ApprovalInitiation	PREPROCESS	PENDING	false	null
14	PostApprovalActions	PREPROCESS	NOTSTARTED	false	null
20	CustomPreProcessHandler	PREPROCESS	NOTSTARTED	true	null
6	CreateUsersActionHandler	ACTION	NOTSTARTED	true	null
13	UserAuditHandler	AUDIT	NOTSTARTED	true	null
19	CreateUserPostProcessHandler	POSTPROCESS	NOTSTARTED	true	null
27	SelfServiceNotificationHandler	POSTPROCESS	NOTSTARTED	true	null
12	PasswordNotificationHandler	POSTPROCESS	NOTSTARTED	true	null
18	PasswordHistoryPostProcessHandler	POSTPROCESS	NOTSTARTED	true	null
24	PostProcessingInitiation	POSTPROCESS	NOTSTARTED	true	null
10	RoleChangeCalculator	POSTPROCESS	NOTSTARTED	true	null
23	PolicyEvaluator	POSTPROCESS	NOTSTARTED	true	null
9	CustomPostProcessHandler	POSTPROCESS	NOTSTARTED	true	null
16	SelfServicePostHandler	POSTPROCESS	NOTSTARTED	true	null
25	RequestCompleted	POSTPROCESS	NOTSTARTED	true	null
11	CreateUserFinalizationHandler	FINALIZATION	NOTSTARTED	false	null

Back To Top

19.5.18 Retry Failed Orchestration

Prerequisite: The following are the prerequisites for running this test:

Prerequisite	Description
Process Id	Enter the Id of the orchestration process.

Description: Obtains the response that indicates how to handle the failure for the given orchestration process.

Result: Displays the orchestration process in failed state and continues to retry based on the response.

19.5.19 SPML Web Service

Prerequisite: None

Description: Verifies that SPML WSDL is accessible and the Web service is up and running.

Result: Displays the contents of SPML WSDL file.

19.5.20 Test OWSM Setup

Prerequisite: None

Description: Verifies OWSM setup by submitting a request with OWSM header information. This also ensures a valid response is returned by submitting a request of listtarget with OWSM header set.

Result: Displays the targets supported by the SPML web-service.

19.5.21 Test SPML to Oracle Identity Manager Request Invocation

Prerequisite: None

Description: SPML WS to Oracle Identity Manager is a signature-based login, This test ensures if this is working, by simulating a Oracle Identity Manager request.

Result: Displays whether signature-based login is working fine.

19.5.22 SPML Attributes to Oracle Identity Manager Attributes

Prerequisite: None

Description: Lists all the mapping of SPML attributes to Oracle Identity Manager attributes which helps the administrator to check if the set up is correct.

Result: Displays a table showing the SPML to Oracle Identity Manager attributes mappings:

SPML Attribute Name	Oracle Identity Manager Attribute Name
Number Format	Number Format
localityName	Locality Name
countryName	Country
manager	User Manager
facsimileTelephoneNumber	Fax

SPML Attribute Name	Oracle Identity Manager Attribute Name
generationQualifier	Generation Qualifier
street	Street
state	State
surname	Last Name
Embedded Help	Embedded Help
Territory	FA Territory
organizationUnit	LDAP Organization Unit
givenName	First Name

19.5.23 Username Test

Prerequisite: None

Description: Lists the existing username generation policy defined in Oracle Identity Manager

Result: Displays the policy name.

19.5.24 Diagnose Creation of User and Role in Oracle Identity Manager and LDAP

Prerequisite: None

Description: Verifies the user creation and role creation are working fine in LDAP and Oracle Identity Manager individually.

Result: Displays the status specifying whether user and role creation was successful in Oracle Identity Manager and LDAP.

19.5.25 Diagnose OVD Connection

Prerequisite: None

Description: Verifies if Oracle Identity Manager is able to connect to the OVD.

Result: Displays whether Oracle Identity Manager was successful to connect to the OVD.

19.5.26 Diagnose LDAP Reserve Container

Prerequisite: None

Description: Oracle Identity Manager configuration file has the tree structure of reserve container. This test validates that the reserve container was created during the setup.

Result: Displays whether reserve container is created properly.

20

Installing and Configuring a Remote Manager

This chapter describes the configuration of Oracle Identity Manager and installation of the Remote Manager. It contains the following topics:

20.1 Overview of Oracle Identity Manager Configuration

To construct adapter tasks, ensure that Oracle Identity Manager has access to the target API JAR files and third-party applications to which you want to connect.

When your adapter uses Java tasks, you must configure Oracle Identity Manager to find the appropriate Java APIs. To do this, you must place the .jar files that contain these APIs into the JavaTasks subdirectory of the *OIM_HOME* folder path, such as C:\oracle\Xellerate\JavaTasks. Then, you can access the Java classes associated with these Java APIs and use them in the Java task you are creating.

Sometimes, instead of directly communicating with the third-party system, Oracle Identity Manager must use an Oracle Identity Manager component that acts like a proxy. This component is known as Remote Manager.

The Remote Manager is used for:

- Invoking nonremotable APIs through Oracle Identity Manager
- Invoking APIs that do not support Secure Sockets Layer (SSL) over secure connections

The procedures in the following sections show you how to:

- Configure Oracle Identity Manager to reference JAR and class files for Java tasks.
- Configure the Remote Manager.

See Also:

"Creating a Java Task" in the *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager* for more information about Java tasks

20.2 Configuring Oracle Identity Manager to Reference JAR and Class Files

To configure Oracle Identity Manager to reference JAR and class files:

1. Open the JavaTasks subdirectory, which can be found within the *OIM_HOME* folder path. For example, C:\oracle\Xellerate\JavaTasks.
2. Place the JAR file or files into this subdirectory. You can then use these files to create Java tasks within an adapter without restarting the server.

20.3 Installing the Remote Manager

To configure the Remote Manager for the application server that you use, follow the instructions described in Oracle Identity Manager installation guide.

20.4 Creating and Testing a Remote Manager IT Resource

This section describes the tasks for creating and testing a Remote Manager IT Resource. It contains the following topics:

Remote Manager is an Oracle Identity Manager component that acts like a proxy in directly communicating with a third-party system. The Remote Manager is used to invoke nonremotable APIs through Oracle Identity Manager that support Secure Sockets Layer (SSL) over secure connections.

After installing the Remote Manager and establishing the trust relation between the Oracle Identity Manager Server and the Remote Manager (trusting the certificate), you can create an IT Resource for the Remote Manager and then test it.

20.4.1 Adding the Trust Relation

After installing the Remote Manager, you can ensure that the certificate is trusted between the application server and the Remote Manager. To do so, first open the Remote Manager form in the Administration folder of Oracle Identity Manager Design Console. The Remote Manager form shows all Remote Managers that are connected but not necessarily "trusted".

Perform the following steps to ensure that the trust relation between the application server and the Remote Manager is established through the certificate. In this procedure, the JBoss Application Server is used as an example. The keytool utility is used to import/export the certificates.

1. Using a command prompt, open the *XLREMOTE_HOME* directory and use the keytool utility to list the certificate fingerprints.
2. Enter the command:

```
$JAVA_HOME/jre/bin/keytool -list -keystore ./config/default-keystore.jks
```

Note: The Oracle Identity Manager keystore is .xlkeystore. It is called default-keystore.jks. It is stored in $*DOMAIN_HOME*/config/fmwconfig/.

For the remote manager, the keystore is stored in $*XLREMOTE_HOME*/config/ directory. The keystore name is default-keystore.jks.

3. Enter the default password for xellerate keystore: KEYSTORE_PASSWORD

   Your keystore contains 1 entry

   xell, Jan 7, 2005, keyEntry,

   Certificate fingerprint (MD5):

   B0:F2:33:C8:69:E4:25:A3:CB:59:E8:51:27:EE:5C:52

   The certificate fingerprint is marked in bold. Compare this to the list of certificates in the keystore.

4. Perform the procedure described in the "Trusting the Remote Manager Certificate" section in the installation guide for the application server that you use.

   Tip: If a create user operation from Oracle Identity Administration with the IT resource set to use SSL fails, then import the certificate in jrocket cacerts and Demotrust.jks, and then create the user. To do so, configure SSL by using the following commands:

   1. To import the certificate in jdk, run the following command:

      keytool -import -keystore *ORACLE_HOME*/cacerts -file *CERTIFICATE_PATH*/*CERTIFICATE_NAME* -storepass changeit

      For example:

      keytool import -keystore /home/testoc4j/OIM091231/jrockit_160_14_R27.6.5-32/jre/lib/security/cacerts -alias adcert14thjan

   2. To import the certificate in DemoTrust.jks, which is in the *WEBLOGIC_SERVER*/server/lib directory, run the following command:

      keytool -import -keystore *WEBLOGIC_SERVER*/server/lib/DemoTrust.jks -file *CERTIFICATE_PATH* -storepass DemoTrustKeyStorePassPhrase

      For example:

      keytool -import -keystore /home/testoc4j/OIM091231/wlserver_10.3/server/lib/DemoTrust.jks -file /home/testoc4j/OIM091231/adcert.cer -storepass DemoTrustKeyStorePassPhrase

20.4.2 To Create and Test a Remote Manager IT Resource

To create and test a Remote Manager IT resource, perform the following steps:

Note: Remote Manager does not support non-SSL communication. By default, one-way SSL communication is supported. If you want to enable two-way SSL communication, then change the value of the <RMSecurity><ClientAuth> property to `True` in the following file:

`$REMOTE_MANAGER/config/xlconfig.xml`

1. In Oracle Identity Manager Design Console, open the Resource Object form.
2. Create a resource object. In this example, the following parameters are set:
   - The name is MyObj
   - The option, Order for User is enabled
   - The Type is Application
   - The following check boxes are available:
     - Allowed Multiple
     - Auto Save
     - Self Request Allowed
     - Allow All
     - Auto Launch
3. Create an IT resource type for the resource object. Open the IT Resource Type Definition form. In this example, the following parameters are set:
   - Server Type: MyObjServer.

   Note: While defining the IT Resource Type parameter in Oracle Identity Manager Design Console, you can specify which fields will be encrypted.
4. Create an IT resource for the Remote Manager. In this example, the following parameters are set:
   - The name of the IT Resource is remote.
   - The name of the Type is Remote Manager.

     Ensure that the IT resource has the proper URL and service name, and that the Remote Manager is installed at the location indicated by the URL.

     Note: Check to see if the name itself is not present in the URL. For example, the Remote Manager is composed of the service name and URL, as follows:

     service name: RManager url: rmi://w2kevandanwkstn:12346
5. Create an instance of the MyObjServer IT Resource Type created previously. Open the IT Resource Information Form. In the Remote Manager field, ensure that the Remote Manager created in Step 4 (remote) is selected.

6. After saving the information in the IT Resources Information form, you can provide any additional details required for that IT resource. In this example, the user name and password are entered.
7. Create a JAR file for the following code:

```
package testme;
import java.io.PrintStream;
public class test
{
    public test ()
    {
    }
    public static int addme(int i, int j)
    {
     /*6*/System.out.println(i + "+" + j + "=" + (i + j));
     /*7*/return i + j;
     }
     public static void main(String args[])
     {
      /*  11*/addme(5, 10);
     }
}
```

   This code will be run on the Remote Manager.

8. Copy the JAR file into the *XLREMOTE_HOME*/JavaTasks and *OIM_HOME*/JavaTasks directories.
9. Create an adapter that will be run in the Remote Manager. Open the Adapter Factory form. In this example, the following parameters are set:
   - The Adapter Name is remotetest.
   - The Adapter Type is Process Task.

     For this example, you can create three variables for this adapter (based on example code in the .jar file). Click **Add.** The Java code takes two integers as arguments and the IT resource as the third variable.
10. In the first variable, the following parameters are set:
    - The Variable name is var1.
    - The Variable type is Integer.
    - The Map To option is set to Resolve at Run time.
11. Create the second variable in the same way you did the first. The following parameters are set:
    - The Variable name is var2.
    - The Variable type is Integer.
    - The Map To option is set to Resolve at Run time.
12. Create the third variable for IT Resource. The parameters are set as follows:
    - The Variable name is ITRes.
    - The Variable type is ITResource.
    - The Map To option is set to Resolve at Run time.
    - The Resource Type is MyObjServer.

Note: The Resource Type field must be the same "ITResource Type" created in Step 5 and not Remote Manager.

13. Add a New Remote Java Task. In the Adapter Factory Form, click **Add**. Ensure that the Functional Task option is active. Select the **Remote** option. Click **Continue**.
14. The Object Instance Selection dialog box is displayed. Create a new Object Instance. Ensure that the New Object Instance option is active. Click **Continue**.
15. The Remote window is displayed. In this example, the following parameters are set:
    - The Task Name is remote.
    - The API Source references the .jar file in the JavaTask folder.
    - The Application API is Testme.test.
    - The Constructor is set to 0 public testme.test ().
    - The Method is set to testme.test.addme (int, int).

    After clicking **Save**, the IT Resource is automatically added as an argument. The Application Method Parameters are ready for mapping.
16. Begin mapping the parameters by highlighting the first item in the Parameter Data Mapping list. This output parameter is an integer. The following mapping is set:
    - Map To: Adapter Variables
    - Name: Return variable
17. Click **Set**.
18. Highlight the second parameter to map. This input parameter is an integer. The following mapping is set:
    - Map: Adapter Variables
    - Name: var1
19. Click **Set**.
20. Select the third parameter to map. This input parameter is an integer. The following mapping is set:
    - Map To: Adapter Variables
    - Name: var2
21. Click **Set**.
22. Select the final parameter to map. Map this ITResource to the variable passed as input to the adapter. The following mapping is set:
    - Map To: Adapter Variables
    - Name: ITRes
23. Click **Set**.
24. Click **Set**. Then click **Save**. The Adapter Factory form is displayed.
25. Compile the adapter by clicking **Build**.

To invoke the adapter, you can create a provisioning process that calls this adapter as one task. To do this:

1. Open the Process Definition Form. In this example, the following parameters are set:
   - The Name field is MyObjProv
   - The Type field is Provisioning
   - The Object name is MyObj

     The following check boxes are available:
     - Default Process
     - Auto Pre-populate
     - Auto Save Form
2. Click the **Save** icon. The provisioning tasks automatically appear in the Tasks tab.
3. Click **Add** to create a new task. In this example, the parameters are set:
   - The Task Name field is Call Remote Adapter.
   - The Task Description field explains the task's function.
4. Click the **Save** icon. Then click the **Integration** tab. Next, click **Add** to add an adapter to this task. The Handler Type window is displayed.
5. Enable the **Adapter** option and select the adapter to be executed.
6. Click the **Save** icon. In the Integration tab, the adapter name appears in the Name field. The Status field shows that the Mapping is incomplete. The Adapter Variables pane shows the variables are not mapped.
7. Select the first variable, Adapter return value, then click **Map**. The Edit Data Mapping for Variable window is displayed. The parameters are set to:
   - Data Type: Object
   - Map To: Response Code
8. Select the second variable, var1 then click **Map**. The **Edit Data Mapping for Variable** window appears. The parameters are set to:
   - Data Type: Integer
   - Map To: Literal
   - Qualifier: Integer
   - Literal Value: 10
9. Select the third variable, var2, then click **Map**. The Edit Data Mapping for Variable window is displayed. The parameters are set to:
   - Data Type: set to Integer
   - Map To: Literal
   - Qualifier: Integer
   - Literal Value: 20
10. Select the fourth variable, ITRes, and then click **Map**. The Edit Data Mapping for Variable window is displayed. The parameters are set to:
    - Data Type: IT Resource (MyObjServer)

- Map To: IT Resource
- Qualifier: MyObjServerInstance

11. Click the **Responses** tab of the Editing Task window. Click **Add** to add the possible responses from the adapter. In this example, the only possible response is 30. Set Description to Completed and Status to C.
12. Click the **Task to Object Status Mapping** tab. In the Completed category, set **Object Status** to Provisioned.
13. At this point, you are ready to directly provision a user with the newly created resource to test the execution of the remote task. However, you must first ensure that the Remote Manager is running. Open the Remote Manager Form and verify that the service is available.
14. Start the Oracle Identity Manager Administrative and User Console and login as Administrator. Navigate to **Users, Manage** and select a user to provision this resource *(MyObj)*. The User Detail page appears with the selected user. In the View Additional Details About This User pull-down option, select **Resource Profile**.
15. The User Detail, Resource Profile page is displayed. Click **Provision New Resource** and select the newly created resource (*MyObj*).
16. The Provision Resource to User wizard is displayed. Click **Continue** to complete the provisioning process.
17. Continue with the provisioning process until you come to the **Resource Successfully Provisioned** page is displayed.
18. Check the Remote Manager log file to see if the code is executed. The Remote Manager log file is located in the *OIM_HOME*/xlremote/log directory. The last line in the log should be similar to the following:

    ```
    DONE5+10=15
    ```

    The preceding line shows that the two input integers are added to equal 15. This indicates that the code executed correctly and that the resource object was provisioned.

20.5 Updating xlconfig.xml file to Change the Port for Remote Manager

To update the xlconfig.xml file and start the remote manager on a new port as opposed to what was set during installation:

1. Access xlconfig.xml from the following path:

   ORACLE_HOME/remote_manager/config/xlconfig.xml
2. Edit the following tags:
   a. ListenPort under RMSecurity for Remote Manager SSL Listen port.
   b. RMIRegistryPort under RMSecurity for RMI Registry.
3. Change the port numbers.
4. Restart the remote manager.

Note:

- You need not perform this procedure during installation. This is required, in case you need to change ports while using the product.
- You must change the port number in the IT resource pointing to the remote manager.

20.6 Configuring the Remote Manager by Using Your Own Certificate

To configure the Remote Manager by using your own certificate on the Remote Manager server:

Note: Perform the procedure given in this section only if you want to use your own certificate instead of the default Oracle Identity Manager certificates. Otherwise, skip this section.

1. Generate a new custom keystore and certificate. Note the password (new_keystore_pwd) that you use for the new keystore.
2. Copy the new keystore to the *OIM_RM_HOME*/config/ directory.
3. In a text editor, open the *OIM_RM_HOME*/config/xlconfig.xml file.
4. Locate the <RMSecurity> tag and change the value in the <Location> and <Password> tags as follows:
   - If you are using the IBM JRE, then change the values to:

```
<KeyStore>
    <Location>new_keystore_name</Location>
    <Password encrypted="false">new_keystore_pwd</Password>
    <Type>JKS</Type>
    <Provider>com.ibm.crypto.provider.IBMJCE</Provider>
</KeyStore>
```

   - For all other JREs, change the values to:

```
<KeyStore>
    <Location>new_keystore_name</Location>
    <Password encrypted="false">new_keystore_pwd</Password>
    <Type>JKS</Type>
    <Provider>sun.security.provider.Sun</Provider>
</KeyStore>
```

5. Restart the Remote Manager server, and open the xlconfig.xml file to ensure that the password for the new keystore is encrypted.

To configure the Remote Manager by using your own certificate on Oracle Identity Manager:

1. Export the certificate from the newly created keystore on Remote Manager server.
2. Copy the new certificate file to the *OIM_DOMAIN_HOME*/config/fmwconfig/ directory.
3. Import certificate into default-keystore.jks.

4. Check if the connection between Remote Manager and Oracle Identity Manager is established.

21

Using the Form Version Control Utility

This chapter describes the scope, content, and description of the Form Version Control (FVC) utility. It contains the following topics:

21.1 FVC Utility Scope

The following table provides a scope of the functions that are implemented with this utility:

Functionality	Implemented (Yes/No)	Comments
Upgrade process form version	Yes	Ensure that the target form version exists and is the active form version.
Upgrade child form version	Yes	The child form version is automatically upgraded to the child form attached with the active parent form.
Update values on parent form	Yes	Ensure that the target form version exists and has the fields whose values you are trying to update.
Update values on child form	Yes	Ensure that the target child form exists and the user is provisioned with the child form.
Insert values on child form	Yes	Ensure that fields that you are inserting exist on the child form version that is attached with the active parent form.

21.2 FVC Utility Content

The following table lists and describes the names and paths of the files that comprise the utility.

File Name with Path	Description
OIM_DC_HOME\xlclient\lib\xlFvcUtil.jar	This JAR file contains the Form Version Control utility classes required to run it.
OIM_DC_HOME\xlclient\xlFvcUtil.ear	This EAR file contains the Form Version Control utility classes required to run it.

File Name with Path	Description
OIM_DC_HOME\xlclient\fvc.properties	This file contains all the configuration properties regarding the source and target form versions, the fields on them, their values, and child form information.
OIM_DC_HOME\xlclient\fvcutil.cmd *OIM_DC_HOME*\xlclient\fvcutil_websphere.cmd	These scripts are used to run the Form Version Control Utility on Microsoft Windows systems. When you run this script, you must provide the Oracle Identity Manager administrator user name and password as shown in the following command: *OIM_DC_HOME*\xlclient\fvcutil.cmd *OIM_ADMINISTRATOR_LOGIN* *OIM_ADMINISTRATOR_PASSWORD*

21.3 FVC Utility Description

The Form Version Control utility is designed to update the version number field of the custom process forms and data in the additional process form fields. The utility is started from the command console, and operates by using command-line parameters to login and a properties file. The parameters in the properties file and validity of user's login and password are verified and appropriate error messages are produced to signify an error when one occurs.

21.4 FVC Utility Features

The following list summarizes the FVC utility:

- Per system requirements, the utility will update only process forms for objects whose status is not `Revoked`.
- The utility has special provisioning for the case where form field values must be updated, but the form version should remain the same. In this case, the *`version to`* and *`version from`* parameters must be the same. The utility will not create an error, but it will update field values for the version specified without changing the version value itself.
- The utility does not have any feature that will allow it to insert a child record. A child table record is considered to be a single child table field. Therefore, if the following entries exist in the `fvc.properties` file, it will create three different rows in the child table, instead of creating and inserting a single child record with the specified values for the three fields:

  ```
  Child;UD_CF3_FIELD7;tiger;Insert

  Child;UD_CF3_FIELD8;mad;Insert

  Child;UD_CF3_FIELD9;me2;Insert
  ```

- The utility can only be used to update custom process forms when a value of Active Version is assigned to the `ToVersion` property in the fvc.properties file.
- Default values for new fields must be defined in the property files.

22

Using the Archival Utilities

This chapter describes how to use the various archival utilities in the following sections:

- Using the Reconciliation Archival Utility
- Using the Task Archival Utility
- Using the Platform Archival Utility
- Using the Requests Archival Utility

22.1 Using the Reconciliation Archival Utility

This section describes how to use the Reconciliation Archival utility. It contains the following topics:

- Understanding the Reconciliation Archival Utility
- Prerequisite for Running the Reconciliation Archival Utility
- Archival Criteria
- Running the Reconciliation Archival Utility
- Log File Generated by the Reconciliation Archival Utility

22.1.1 Understanding the Reconciliation Archival Utility

Oracle Identity Manager stores reconciliation data from target systems in Oracle Identity Manager tables called **active reconciliation tables**:

During the reconciliation process, Reconciliation Manager reconciles data in the active reconciliation tables with the Oracle Identity Manager core tables. Because Reconciliation Manager does not remove reconciled data from the active reconciliation tables, they might eventually grow very large, resulting in decreased performance during the reconciliation process. You can use the Reconciliation Archival utility to archive data that has been reconciled with Oracle Identity Manager. The Reconciliation Archival utility stores archived data in the **archive reconciliation tables**, which have the same structure as the active reconciliation tables.

Table 22–1 lists the active reconciliation tables with the corresponding archive reconciliation tables in which data from the active reconciliation tables are archived.

Table 22–1 Active and Archive Reconciliation Tables

Active Reconciliation Tables (Oracle Identity Manager Tables)	Archive Reconciliation Tables
RECON_EVENTS	ARCH_RECON_EVENTS
RECON_JOBS	ARCH_RECON_JOBS
RECON_BATCHES	ARCH_RECON_BATCHES
RECON_EVENT_ASSIGNMENT	ARCH_RECON_EVENT_ASSIGNMENT
RECON_EXCEPTIONS	ARCH_RECON_EXCEPTIONS
RECON_HISTORY	ARCH_RECON_HISTORY
RECON_USER_MATCH	ARCH_RECON_USER_MATCH
RECON_ACCOUNT_MATCH	ARCH_RECON_ACCOUNT_MATCH
RECON_CHILD_MATCH	ARCH_RECON_CHILD_MATCH
RECON_ORG_MATCH	ARCH_RECON_ORG_MATCH
RECON_ROLE_MATCH	ARCH_RECON_ROLE_MATCH
RECON_ROLE_HIERARCHY_MATCH	ARCH_RECON_ROLE_HIER_MATCH
RECON_ROLE_MEMBER_MATCH	ARCH_RECON_ROLE_MEMBER_MATCH
RA_LDAPUSER	ARCH_RA_LDAPUSER
RA_MLS_LDAPUSER	ARCH_RA_MLS_LDAPUSER
RA_LDAPROLE	ARCH_RA_LDAPROLE
RA_MLS_LDAPROLE	ARCH_RA_MLS_LDAPROLE
RA_LDAPROLEMEMBERSHIP	ARCH_RA_LDAPROLEMEMBERSHIP
RA_LDAPROLEHIERARCHY	ARCH_RA_LDAPROLEHIERARCHY
All reconciliation horizontal tables	"ARCH_" + substr(HTnames,1,25)

You can use the Reconciliation Archival utility to perform the following tasks:

- Archive all or specific data from the active reconciliation tables to the archive reconciliation tables
- Delete all data from the active reconciliation tables

When you archive data by moving it from the active reconciliation tables to the archive reconciliation tables, you must specify the date in the YYYYMMDD format, such as all records before this date will be archived, and a reconciliation event status parameter value, which defines the data that you want to archive. For information about these archiving criteria, refer to "Archival Criteria" on page 3.

If you choose to archive selective data, then the utility archives data that falls in the specified date range and event status.

When you archive all data from the active reconciliation tables to the archive reconciliation tables, the Reconciliation Archival utility archives all reconciliation data with event status of Event Linked or Event Closed.

The files that constitute the Oracle Database version of the Reconciliation Archival utility are located in the following directory:

OIM_HOME/db/oim/oracle/Utilities/Recon11gArchival

22.1.2 Prerequisite for Running the Reconciliation Archival Utility

Before running the Reconciliation Archival utility, the OIM_RECON_ARCH tablespace must be created in the database. To do so, you can run the following sample command:

```
CREATE TABLESPACE OIM_RECON_ARCH
        LOGGING DATAFILE 'OIM_RECON_ARCH'
        SIZE 500M REUSE AUTOEXTEND ON NEXT 10M;
```

Note:

- You must set *LD_LIBRARY_PATH* to start Oracle utilities such as SQL*Plus in the environment where you want to run Oracle Identity Manager utilities.
- Data that has been archived from the active reconciliation tables to the archive reconciliation tables will no longer be available through Oracle Identity Manager. To access this data, you must query the archive reconciliation tables in your Oracle Identity Manager database.

22.1.3 Archival Criteria

To select reconciliation data to archive, provide the following criteria. Data with matching values will be archived.

- Date must be in the format YYYYMMDD. All records before this date that match the specified reconciliation event parameter value will be archived.
- Select Closed, Linked, Closed or Linked, or All for the reconciliation event parameter.
  - Closed describes events that have been manually closed in Reconciliation Manager.
  - Linked describes events that were reconciled in Oracle Identity Manager, including the following states:
    * Creation Succeeded
    * Update Succeeded
    * Delete Succeeded
    * Creation Failed
    * Update Failed
    * Delete Failed
  - Closed or Linked
  - All archives all events regardless of status

22.1.4 Running the Reconciliation Archival Utility

To run the Reconciliation Archival utility:

1. Ensure that the Oracle Identity Manager database is available and that no reconciliation processes are running. In addition, ensure that the Oracle Identity Manager database is not open to transactions for other sessions.

Note: Oracle recommends that you run the Reconciliation Archival utility during off-peak hours.

2. Stop the Oracle Identity Manager by following the instructions in the "Starting and Stopping Servers" chapter.
3. On Microsoft Windows platforms, you must specify the short date format as M/d/yyyy. In addition, you must specify the time format as H:mm:ss. To customize the date and time formats, use the Regional and Language Options command in Control Panel.

   Note:

   - When you change the date and time format, the change is applied to all the applications running on the Microsoft Windows platform.
   - Minimal validation is done on date before calling the utility, and you can scan logs files for any ORA-18xx errors for invalid date-related errors.

4. On Linux or UNIX platforms, run the following commands to set execution permission for the oim_recon_archival.sh file and to ensure that the file is a valid Linux or UNIX text file:

   ```
   chmod 755 path/oim_recon_archival.sh
   dos2unix path/oim_recon_archival.sh
   ```

5. On Linux or UNIX platforms, run the *path*/oim_recon_archival.sh file. On Microsoft Windows platforms, run the *path*\oim_recon_archival.bat file.
6. For Oracle Database installations, enter values for the following parameters when prompted:
   - Oracle home directory
   - Input for validating whether the Oracle Identity Manager database is running on a remote computer
   - For a remote database, a connection string is required as input, which is of the following format: //HOST_NAME:PORT/SERVICE_NAME
   - Oracle Identity Manager database user name and password
7. When prompted, select a reconciliation event status for the data that you want to archive:
   - Enter 1 for Closed
   - Enter 2 for Linked
   - Enter 3 for Closed or Linked
   - Enter 4 for All
   - Enter 5 for Exit
8. Enter the reconciliation creation date in the YYYYMMDD format. All records before this date with required status value will be archived.
9. Enter the batch size for processing.

The default batch size is 2000.

10. On Microsoft Windows platforms, reset the short date format to the date format for your region or locale after you run the utility. Use the Regional and Language Options command in Control Panel to reset the date format.
11. Because the data from active reconciliation tables are removed, your DBA must analyze the active reconciliation tables and their indexes in order to update the statistics. Perform this step only if you are using Oracle Database as the database for Oracle Identity Manager.

22.1.5 Log File Generated by the Reconciliation Archival Utility

After running the Reconciliation Archival utility, the following log file is generated:

./logs/oim_recon_archival_summary_*TIMESTAMP*.log

If the running the utility fails, then the log file records the batch number at which the utility fails along with the error messages.

22.2 Using the Task Archival Utility

This section describes how to use the Task Archival utility. It contains the following topics:

- Understanding the Task Archival Utility
- Preparing Oracle Database for the Task Archival Utility
- Running the Task Archival Utility
- Reviewing the Output Files Generated by the Task Archival Utility

22.2.1 Understanding the Task Archival Utility

In Oracle Identity Manager, a **task** refers to one or more activities that comprise a process, which handles the provisioning of a resource. For example, a process for requesting access to a resource may include multiple provisioning tasks. Oracle Identity Manager stores task data in the following tables, which are called **active task tables**:

- OSI
- OSH
- SCH

By default, Oracle Identity Manager does not remove completed tasks from the active task tables. As the size of the active task tables increases, you might experience a reduction in performance, especially when managing provisioning tasks. After a task executes successfully, you can use the Task Archival utility to archive the task data and remove it from the active task tables. Archiving task data with the Task Archival utility improves performance and ensures that the data is safely stored.

The Task Archival utility stores archived task data in the following **archive task tables**, which have the same structure as the active task tables:

- ARCH_OSI
- ARCH_OSH
- ARCH_SCH

You can use the Task Archival utility to archive the following types of tasks:

- Provisioning tasks for resource instances that have been revoked for disabled or deleted users
- Provisioning tasks for resource instances that have been revoked

When you archive tasks with the Task Archival utility, you can specify the type of archive operation, the user status, the task execution date, and the number of records above which to drop the indexes before archiving. The archive operation represents the type of task data to archive and the user status determines whether to archive data for users who have been deleted, disabled, or both. The task execution date represents the date on which a task is executed and must be in the format YYYYMMDD.

All executed tasks, up to the task execution date you specify, will be archived. To reduce the time that the archiving process takes, the utility drops the indexes on all active task tables when the number of records to be archived is greater than 200000. The indexes are re-created after the archived data is deleted from the active task tables. You can change the value 200000 to your preferred value. You can change the value in the following lines of code in the OIM_TasksArch.bat file or in the OIM_TasksArch.sh file:

In the .bat file, set INDXRESP=200000

In the .sh file, indxopt=200000

The files that constitute the Oracle Database version of the Task Archival utility are located in the following directory:

OIM_HOME/db/oim/oracle/Utilities/TaskArchival

Note: Data that has been archived from the active task tables to the archive task tables will no longer be available through Oracle Identity Manager. To access this data, you must query the archive task tables in your Oracle Identity Manager database.

22.2.2 Preparing Oracle Database for the Task Archival Utility

Before you can use the Task Archival utility with Oracle Database, you must perform the following steps:

1. Start SQL*Plus and connect to Oracle Database as a SYS user.
2. Create a separate tablespace for the archival task tables by entering the following command. Replace *DATA_DIR* with the directory in which you want to store the data file and adjust the size and other parameters as necessary for your environment.

```
CREATE TABLESPACE TasksArch
    DATAFILE 'DATA_DIR\tasksarch_01.dbf' SIZE 1000M REUSE
    EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

Note: Oracle recommends that you allocate a large UNDO tablespace when archiving large amounts of data. In addition, turn on parallel execution by configuring the parallel_max_servers and parallel_min_servers initialization parameters. Parallel execution helps improve the performance of the archival process.

3. Connect to Oracle Database as the Oracle Identity Manager database user.
4. Enter the following command to run the cr_taskarchival_ddl_table.sql script, which creates a table named OIM_TASK_ARCH_DDL. This table is used by the Task Archival utility.

   ```
   @ path/cr_taskarchival_ddl_table.sql
   ```

5. Enter the following command to run the Create_TasksArch_Tables.sql script, which creates the archive task tables:

   ```
   @ path/Create_TasksArch_Tables.sql
   ```

6. Enter the following command to run the OIM_SP_TASKS_ARCHIVAL.sql script, which creates a stored procedure that the Task Archival utility uses to archive and delete task data:

   ```
   @ path/OIM_SP_TASKS_ARCHIVAL.sql
   ```

7. If your Oracle Database instance is running in ARCHIVELOG mode, you must switch to NOARCHIVELOG mode before running the Task Archival utility. See *Oracle Database Administrator's Guide* for information about changing the database archiving mode.

   Note: You must set *LD_LIBRARY_PATH* to start Oracle utilities such as SQL*Plus in the environment where you want to run Oracle Identity Manager utilities.

22.2.3 Running the Task Archival Utility

Perform the following steps to run the Task Archival utility:

1. Ensure that the Oracle Identity Manager database is available and that no reconciliation processes are running. Also, ensure that the Oracle Identity Manager database is not open to transactions for other sessions.

   Note: Oracle recommends that you run the Task Archival utility during off-peak hours.

2. Back up the OSI, SCH, and OSH tables.
3. Stop Oracle Identity Manager by following the instructions in the Oracle Identity Manager installation guide for your application server.
4. On Microsoft Windows platforms, you must specify the short date format as dddd M/d/yyyy. In addition, you must specify the time format as H:mm:ss. To customize the date and time formats, select the Regional and Language Options command in the Control Panel.

Note:

- When you change the date and time format, the change is applied to all the applications running on the Microsoft Windows platform
- Minimal validation is done on date before calling the utility, and you can scan logs files for any ORA-18xx errors for invalid date-related errors

5. On Linux and UNIX platforms, run the path/OIM_TasksArch.sh file. On Microsoft Windows platforms, run the path`\OIM_TasksArch.bat` file.
6. For Oracle Database installations, enter values for the following parameters when prompted:
   - Oracle home directory
   - Oracle Identity Manager database name or TNS string if the Oracle Identity Manager database is running on a remote computer
   - For a remote database, a connection string is required as input, which is of the following format: //HOST_NAME:PORT/SERVICE_NAME
   - Oracle Identity Manager database user name and password
7. When prompted, select one of the following options:
   - Archive all provisioning tasks on resource instances that have been revoked for disabled or deleted users.
   - Archive all provisioning tasks on resource instances that have been revoked.
   - Exit.
8. If you chose to archive all provisioning tasks for resource instances that have been revoked for disabled or deleted users, select one of the following options:
   - Users at Deleted status
   - Users at Disabled status
   - Users at Deleted and Disabled status
   - Go back to Main Menu
9. Enter a task execution date in the format YYYYMMDD when prompted. All executed tasks, up to the task execution date you specify, will be archived. To archive all tasks that were executed on or before the current date, press **Enter** without entering a date.
10. Enter a value of **y** or **Y** when prompted to archive the tasks. Otherwise, enter a value of **n** or **N** to exit the utility.

    Note: You must enter the value of Y or N when prompted. If you press Enter without selecting a value, then the utility again counts the number of tasks to be archived and prompts you without beginning the archive.

11. On Microsoft Windows platforms, reset the short date format to the date format for your region or locale after the Task Archival utility finishes running. Use the

Regional and Language Options command in the Control Panel to reset the date format.

Note: You must analyze the active task tables and their indexes for updated statistics, because the data from active task tables is removed. Perform this step only if you are using Oracle Database as the database for Oracle Identity Manager.

22.2.4 Reviewing the Output Files Generated by the Task Archival Utility

Table 22–2 describes the output files that are generated by the Task Archival utility.

Table 22–2 Output Files Generated by the Task Archival Utility

File	Description
`Err_DB_Conn_`*`timestamp`*`.log`	Generated when the utility is unable to connect to the database with the specified credentials
`Err_Arch_Tasks_`*`timestamp`*`.log`	Generated when the archival or deletion processes fail
`Arch_TaskData_`*`timestamp`*`.log`	Generated when the archival or deletion processes succeed

Note: These error log files are deleted when you run the utility again.

22.3 Using the Platform Archival Utility

This section describes how to use the Platform Archival utility. It contains the following topics:

- What is Platform Archival Utility?
- Scripts Constituting the Platform Archival Utility
- Preparing Oracle Database for the Platform Archival Utility
- Running the Platform Archival Utility
- Platform Archival Utility Menu Options
- Output Files Generated by the Platform Archival Utility

22.3.1 What is Platform Archival Utility?

Oracle Identity Manager stores platform data from the target systems in the following tables, which are called **active platform tables**:

- ORCHPROCESS
- ORCHEVENTS
- ORCHFAILEDEVENTS
- CONTEXT
- CONTEXTVALUE

By default, Oracle Identity Manager does not remove completed transaction data from the active platform tables. As the size of the active platform table increases, you might

experience a deterioration in the performance of Oracle Identity Manager. You can use the Platform Archival utility to archive data processed by Oracle Identity Manager and remove it from the active platform table. Archiving transaction data with the Platform Archival utility enhances performance and ensures that the data is safely stored.

> **Note:** Platform data cleanup can also be performed by using the Orchestration Process Cleanup Task scheduled task. However, this task does not support archival of orchestration process data. See "Predefined Scheduled Tasks" on page 2-4 for information about this scheduled task.

The Platform Archival utility stores archived data in the following tables, called **archival platform tables**, which have the same structure as the active platform tables:

- ARCH_ORCHPROCESS
- ARCH_ORCHEVENTS
- ARCH_ORCHFAILEDEVENTS
- ARCH_CONTEXT
- ARCH_CONTEXTVALUE

> **Note:** Data archived from the active platform tables to the archival platform tables is not available through Oracle Identity Manager. You must query the archival platform tables in your Oracle Identity Manager database to access the required data.

The files that constitute the Oracle Database version of the Platform Archival utility are located in the following directory:

OIM_HOME/db/oim/oracle/Utilities/PlatformArchival

22.3.2 Scripts Constituting the Platform Archival Utility

Table 22–3 lists the files that comprise the Platform Archival Utility.

Table 22–3 Scripts Constituting the Platform Archival Utility

Script	Description
OIM_PlatformArch.bat	It is a driver batch file used in Microsoft Windows environment for Platform Archival Utility.
OIM_PlatformArch.sh	It is a driver Shell Script used in UNIX environment for Platform Archival Utility.
OIM_SP_PlatformArchival	It is a stored procedure used for archiving data.
Create_PlatformArch_Tables.sql	It is a script that is used to create the archival platform tables.
Cr_PlatformArchival_DDL_Table.sql	It is a script to create a table to store the DDL commands for recreating the indexes and constraints.

22.3.3 Preparing Oracle Database for the Platform Archival Utility

Before you can use the Platform Archival utility with Oracle Database, you must perform the following steps:

1. Start Oracle SQL*Plus and connect to Oracle Database as a user with Create Tablespace privileges or SYSDBA privileges, such as `SYS` user.
2. Enter the following command to run the oim_archival_tablespace_setup.sql script, which creates a separate tablespace for the archival platform tables:

```
@ path/oim_archival_tablespace_setup.sql
```

> **Note:** You must set *LD_LIBRARY_PATH* to start Oracle utilities such as SQL*Plus in the environment where you want to run Oracle Identity Manager utilities.

22.3.4 Running the Platform Archival Utility

Perform the following steps to run the Platform Archival utility:

1. Ensure that the Oracle Identity Manager database is available. In addition, verify that the Oracle Identity Manager database is not open for transactions with other sessions.

> **Note:** Oracle recommends that you run the Platform Archival utility during off-peak hours.

2. Back up the `ORCHPROCESS`, `ORCHEVENTS`, `ORCHFAILEDEVENTS`, `CONTEXT`, and `CONTEXTVALUE` tables.
3. Stop the Oracle Identity Manager by following the instructions in the "Starting and Stopping Servers"chapter.
4. On Microsoft Windows platforms, specify the short date format as `dddd M/d/yyyy`. In addition, you must specify the time format as `H:mm:ss`.

   To customize the date and time format, select the **Regional and Language Options** command in the Control Panel.

> **Note:**
>
> - When you change the date and time format, the change is applied to all the applications running on the Microsoft Windows platform.
> - Minimal validation is done on date before calling the utility, and you can scan logs files for any ORA-18xx errors for invalid date-related errors.

5. On Linux and UNIX platforms, run the following commands to set execution permission for the `OIM_PlatformArch.sh` file and to ensure that the file is a valid Linux and UNIX text file:

```
chmod 755 path/OIM_PlatformArch.sh
dos2unix path/OIM_PlatformArch.sh
```

6. On Linux and UNIX platforms, run the `path/OIM_PlatformArch.sh` file. On Microsoft Windows platforms, run the `path\OIM_PlatformArch.bat` file.
7. For Oracle Database installations, enter values for the following parameters when prompted:
   - Oracle home directory
   - Oracle Identity Manager database name or TNS string, if the Oracle Identity Manager database is running on a remote computer
   - For a remote database, a connection string is required as input, which is of the following format: //HOST_NAME:PORT/SERVICE_NAME
   - Oracle Identity Manager database user name and password
8. Enter the date range in `YYYYMMDD` format when prompted.
9. When prompted, select one of the following menu options:
   - 1) Archive Orchestration Process Instance Data
   - 2) Archive Context Data
   - 3) Exit

   See "Platform Archival Utility Menu Options" on page 22-12 for more information.
10. If you select `Archive Orchestration Process Instance Data`, you are prompted for the archival criteria.
11. Enter the batch size for processing.

    The default batch size is 2000.
12. The count of records to be archived is displayed. Enter `y` or `Y` when prompted to archive the data. Alternatively, enter a value of `n` or `N` to exit the utility.

    Note: You must enter the value of `Y` or `N` when prompted. If you press Enter without selecting a value, then the utility again counts the number of records to be archived and prompts you without beginning the archival process.

13. On Microsoft Windows platform, reset the short date format to the date format of your region or locale after you run the utility. Select the **Regional and Language Options** command in Control Panel to reset the date format.

    Note: You must analyze the active tables and their indexes for updated statistics, because the data from active tables is removed. You need to perform this step, if and only if you are using Oracle Database as the database for Oracle Identity Manager.

22.3.5 Platform Archival Utility Menu Options

The Platform Archival Utility presents the following menu options.

- Archive Orchestration Process Instance Data
- Archive Context Data

22.3.5.1 Archive Orchestration Process Instance Data

This option archives the orchestration process instance data from ORCHPROCESS, ORCHEVENTS, and ORCHFAILEDEVENTS tables into ARCH_ORCHPROCESS, ARCH_ORCHEVENTS, and ARCH_ ORCHFAILEDEVENTS tables based on the archival criteria. The archival criteria comprise two columns MODIFIEDON and STATUS.

The format for the MODIFIEDON archival criteria is YYYYMMDD and it is mapped to the MODIFIEDON column in the ORCHPROCESS table. The MODIFIEDON date is validated and an error is raised, if it is invalid.

The STATUS criterion is mapped to the STATUS column in the ORCHPROCESS table. The STATUS criterion is COMPLETED and is fixed.

22.3.5.2 Archive Context Data

This option archives the context data from CONTEXT and CONTEXTVALUE tables into ARCH_CONTEXT and ARCH_CONTEXTVALUE tables based on the archival criteria.

While archiving the utility excludes the context ids that are present in the REQUEST, FAILED_TASKS, and ORCHPROCESS tables with STATUS as PENDING.

The format for the MODIFIEDON archival criterion is YYYYMMDD and it is mapped to the MODIFIEDON column in the CONTEXT table. The archival criterion is validated and an error is raised, if it is invalid. In case the MODIFIEDON date selected for archival is less than three months, then the following warning is issued:

```
"WARNING: You are archiving data less than 3 months old. Any pending transaction
information will be lost, and those transactions cannot be completed. Do you want
to continue?".
```

22.3.6 Output Files Generated by the Platform Archival Utility

Table 22–4 describes the output files that are generated by the Platform Archival utility.

Table 22–4 Output Files Generated by the Platform Archival Utility

File	Description
Err_DB_Conn_timestamp.log	Generated when the utility is unable to connect to the database with the provided credentials
Err_Arch_Platform_timestamp.log	Generated when the archival processes fail
Arch_Platform_timestamp.log	Generated when the archival processes succeed

> **Note:** These error log files are deleted when you run the utility again.

22.4 Using the Requests Archival Utility

This section describes how to use the Requests Archival utility. It contains the following topics:

- Understanding the Requests Archival Utility
- Prerequisites for Running the Requests Archival Utility
- Input Parameters

- Running the Requests Archival Utility
- Log Files Generated by the Utility

22.4.1 Understanding the Requests Archival Utility

Requests that are closed or withdrawn are stored in the Oracle Identity Manager database. To archive these requests and free up the disk space and thereby enhance database performance, the Requests Archival utility is used. You can archive request data based on request creation date and request status. Archiving requests based on the request status is optional. By using request status, you can archive:

- Completed requests such as requests with status Withdrawn, Closed, and Completed. This is specified by the user input `1`.
- Completed and failed requests such as requests with status Withdrawn, Closed, Completed, Failed, and Partially Failed. This is specified by the user input `2`.
- All requests based on request creation date. This is specified by the user input `3`.

The Requests Archival utility involves running the following scripts:

- oim_request_archival.bat and oim_request_archival.sh: Driver script batch for Microsoft Windows and shell script for UNIX
- oim_create_request_arch_tables.sql: Script with PL/SQL procedure to create archival tables against all tables that need archiving
- oim_request_archival.sql: Script with PL/SQL procedure that performs the archive

Table 22–5 lists the names of the tables which are to be archived and the corresponding archival table names.

Table 22–5 Archival Tables

Main Table	Archival Table
REQUEST	REQUEST_ARCH
REQUEST_HISTORY	ARCH_REQUEST_HISTORY
REQUEST_APPROVALS	ARCH_REQUEST_APPROVALS
REQUEST_ENTITIES	ARCH_REQUEST_ENTITIES
REQUEST_ENTITY_DATA	ARCH_REQUEST_ENTITY_DATA
REQUEST_BENEFICIARY	ARCH_REQUEST_BENEFICIARY
REQUEST_BENEFICIARY_ENTITIES	ARCH_REQUEST_BE
REQUEST_BENEFICIARY_ENTITYDATA	ARCH_REQUEST_BED
REQUEST_TEMPLATE_ATTRIBUTES	ARCH_REQUEST_TA
WF_INSTANCE	ARCH_WF_INSTANCE
REQUEST_COMMENTS	ARCH_REQUEST_COMMENTS

The files that constitute the Oracle Database version of the Requests Archival utility are located in the following directory:

OIM_HOME/db/oim/oracle/Utilities/RequestArchival

22.4.2 Prerequisites for Running the Requests Archival Utility

Before running the Requests Archival utility:

> **Note:** You must set *LD_LIBRARY_PATH* to start Oracle utilities such as SQL*Plus in the environment where you want to run Oracle Identity Manager utilities.

- Create the OIM_REQUEST_ARCH tablespace. When the Requests Archival utility is run for the first time, a corresponding archival table is created for all the tables that are to be archived. The archival tables are created in a separate tablespace named OIM_REQUEST_ARCH. This tablespace must be created before running the utility.
- Create the required archival tables for the request tables by running the oim_create_request_arch_tables.sql script. This is the PL/SQL script to create archival tables against all tables that are to be archived.

22.4.3 Input Parameters

Table 22–6 lists the input parameters used by the Requests Archival utility:

Table 22–6 Input Parameters

Parameter	Description
Oracle Home	The value of *ORACLE_HOME* environment variable on the system.
Oracle SID	The SID of the Oracle Identity Manager database. For a remote database, a connection string is required as input, which is in the following format: *//HOST_NAME:PORT/SERVICE_NAME* Here, *HOST_NAME* is the host name of the computer on which the database is deployed, *PORT* is the port number of the host, and *SERVICE_NAME* is the name of the database instance.
OIM DB User	The database login ID of the Oracle Identity Manager database user.
OIM DB Pwd	The password of the Oracle Identity Manager database user.
Request Status	The request status based on the user inputs 1, 2, or 3.
Request Creation Date	The utility archives all requests created on or before this request creation date with the required request status.
Batch Size	The utility processes a group of records or batch as a single transaction. The batch size can influence the performance of the utility. Default value of Batch Size is `2000`.

22.4.4 Running the Requests Archival Utility

To run the Requests Archival utility:

1. Ensure that the Oracle Identity Manager database is available. In addition, ensure that the Oracle Identity Manager database is not open to transactions for other sessions.

Note: It is recommended that you run the Requests Archival utility during off-peak hours.

2. Stop Oracle Identity Manager by following the instructions in the "Starting and Stopping Servers" chapter.
3. On Microsoft Windows platform, you must specify the short date format as `dddd M/d/yyyy`. In addition, you must specify the time format as `H:mm:ss`. To customize the date and time formats, use the Regional and Language Options command in Control Panel.

   Note:

   - When you change the date and time format, the change is applied to all the applications running on the Microsoft Windows platform.
   - Minimal validation is done on date before calling the utility, and you can scan logs files for any ORA-18xx errors for invalid date-related errors.

4. On UNIX platform, run the following commands to set execution permission for the `OIM_request_archival.sh` file and to ensure that the file is a valid UNIX text file:

   ```
   chmod 755 path/OIM_request_archival.sh
   dos2unix path/OIM_request_archival.sh
   ```

5. On UNIX platform, run the *path*`/OIM_request_archival.sh` file. On Microsoft Windows platform, run the *path*`\OIM_request_archival.bat` file.

   The oim_request_archival script validates the database input and establishes a connection with the database. It then calls the oim_request_archival.sql script, the script is used to compile PL/SQL procedures related to the utility.

6. For Oracle Database installations, enter values for the following parameters when prompted:

   - Oracle home directory.
   - Oracle Identity Manager database name or TNS string if the Oracle Identity Manager database is running on a remote computer. Otherwise, enter ORACLE SID.
   - For a remote database, a connection string is required as input, which is of the following format:

     //HOST_NAME:PORT/SERVICE_NAME

   - Oracle Identity Manager database user name and password.

7. When prompted, enter one of the following options:

   - Enter 1 to archive the requests with status Request Withdrawn, Request Closed, or Request Completed, and requests with creation date on or before the request creation date specified by the user in the format YYYYMMDD.
   - Enter 2 to archive the requests with status Request Withdrawn, Request Closed, Request Completed, or Request Partially Failed, and requests with

creation date on or before the request creation date specified by the user in the format YYYYMMDD.

- Enter 3 to archive all the requests with request creation date on or before the request creation date specified by the user in the format YYYYMMDD.

8. Specify the batch size, when prompted.
9. On Microsoft Windows platforms, reset the short date format to the date format for your region or locale after you run the utility. Use the Regional and Language Options command in Control Panel to reset the date format.
10. Because the data from active request tables are removed, your DBA must analyze the active request tables and their indexes in order to update the statistics. Perform this step only if you are using Oracle Database as the database for Oracle Identity Manager.

22.4.5 Log Files Generated by the Utility

All the logs are written to the logs/ directory created in the current folder. Table 22–7 lists the log files generated by the utility.

Table 22–7 Logs Generated by the DB Archival Utility

Log File	Description
oim_create_request_arch_tables.log	Created when the utility fails to create the archival tables
oim_request_archival.log	Created when the utility fails to create the procedures required for archival
validate_date.log	Created when the input REQUEST_CREATION_DATE is invalid
oim_request_archival_summary_*TIMESTAMP*.log	Contains the summary of the run
Err_DB_Conn_*TIMESTAMP*_*ATTEMPTNUMBER*.log	Created when the utility is unable to connect to the database with the credentials provided

Part V

Performance Tuning and Best Practices

This part describes the performance tuning of various Oracle Identity Manager components.

It contains the following chapters:

23

Tuning Oracle Database

As with any enterprise class business application, there is no simple procedure for tuning that works for all systems. This section describes one sample configuration and outlines the principles for tuning Oracle Database.

Oracle Identity Manager has many configuration options. The best way to identify bottlenecks and optimize performance is to monitor key database performance indicators in your production environment and adjust the configuration accordingly. This chapter serves as a guideline to help you choose the initial baseline database configuration.

This chapter discusses the following topics:

- Using Database Roles/Grants for Oracle Identity Manager Database
- Sample Instance Configuration Parameters
- Physical Data Placement
- Database Performance Monitoring

23.1 Using Database Roles/Grants for Oracle Identity Manager Database

As a database administrator, you can create roles to grant all privileges to a secure application role required to run a database application. You can then grant the secure application role to other roles or users. An application can have various roles, each granted a different set of privileges that allow the user access more or less data while using the application. For example, you can create a role with a password to prevent unauthorized use of the privileges granted to the role. An application can be designed in such a way so that when it starts, it enables the proper role. As a result, an application user does not need to know the password for an application's role.

Depending on what is granted or revoked, a grant or revoke takes effect at different times, such as:

- All grants and revokes for system and object privileges to users, roles, and PUBLIC grants take immediate effect.
- All grants and revokes of roles to users, other roles, and PUBLIC take effect only when a current user session issues a SET ROLE statement to re-enable the role after the grant and revoke, or when a new user session is created after the grant or revoke.

You can see which roles are currently enabled by examining the SESSION_ROLES data dictionary view.

In Oracle Identity Manager, there are prerequisite grants that are provided to Oracle Identity Manager schema to create necessary objects before installing Oracle Identity

Manager. Some of these grants can be revoked later on after installing the Oracle Identity Manager and can be granted to particular users in future as required by the application.

Table 23–1 describes the grants required for database applications:

Table 23–1 Role Grants for Database Applications

Role Name	Description	Usage Specific to Oracle Identity Manager	If Revoked
CREATE TABLE	Enables a user to create, modify, and delete tables in the user's schema.	Although this is part of grant resource, this is explicitly required because the grant resource does not allow to create a table through a procedure.	User will not be able to create any new tables programmatically. You can revoke this grant when the Oracle Identity Manager deployment is stable, which means all the components and connectors are imported and working as expected. This is because each connector creates its own schema object. This grant is needed for initial run of any archival utility because the archival utilities create tables programmatically.
CONNECT	Provides the create session privileges	To create sessions for users	This can be replaced with create session after installation. You can do this when the Oracle Identity Manager deployment is stable, which means all the components and connectors are imported and working as expected. This is because each connector creates its own schema object.

Table 23–1 (Cont.) Role Grants for Database Applications

Role Name	Description	Usage Specific to Oracle Identity Manager	If Revoked
RESOURCE	Enables a user to create, modify, and delete certain types of schema objects in the schema associated with that user. Grant this role only to developers and to other users that must create schema objects. This role grants a subset of the create object system privileges. For example, it grants the CREATE TABLE system privilege, but does not grant the CREATE VIEW system privilege. It grants the following privileges: ■ CREATE CLUSTER ■ CREATE INDEXTYPE ■ CREATE OPERATOR ■ CREATE PROCEDURE ■ CREATE SEQUENCE ■ CREATE TABLE ■ CREATE TRIGGER ■ CREATE TYPE In addition, this role grants the UNLIMITED TABLESPACE system privilege, which effectively assigns a space usage quota of UNLIMITED on all tablespaces in which the user creates schema objects.	To create sequences, indexes, procedures, triggers, and packages	User will not be able to create any database objects. Only SYS user will be able to do so. You can revoke this grant when the Oracle Identity Manager deployment is stable, which means all the components and connectors are imported and working as expected. This is because each connector creates its own schema object. Specify the quota for tablespaces correctly.
CREATE VIEW	Enables a user to create, modify, and delete views in the user's schema	To create SDP_VISIBLE_V, SDP_REQUIRED_V, SDP_LOOKUPCODE_V, and SDP_RECURSIVE_V views in Oracle Identity Manager	The user will not be able to create any views. Only SYS user will be able to do so.
DBMS_SHARED_POOL	Fits a database object in a shared pool memory	Used for pinning all the procedures and functions used in Oracle Identity Manager in shared memory	It can be revoked after installation but may impact performance because some of the procedures and functions may not be pinned explicitly. The pin_obj procedure is created only for Oracle Identity Manager. It is used to explicitly pin database objects into shared memory. Before revoking this role, make sure that the database-level trigger cache_seq is dropped, if already created.

Table 23–1 (Cont.) Role Grants for Database Applications

Role Name	Description	Usage Specific to Oracle Identity Manager	If Revoked
SYS.DBMS_SYSTEM	Enables an XA Resource Manager and sets privileges so that the XA Resource Manager can manage the interaction between the Oracle database and the applications. **Note:** Each database connection is enlisted with the transaction manager as a transactional resource. The transaction manager obtains an XA Resource for each connection participating in a global transaction. The transaction manager uses the start method to associate the global transaction with the resource, and it uses the end method to disassociate the transaction from the resource. The resource manager associates the global transaction to all work performed on its data between the start and end method invocations.	For XA resource and database transactions	On Oracle Database version 10.2.0.4 onwards, it can be removed safely. Oracle has redeemed themselves by moving the DIST_TXN_SYNC procedure to a new package called DBMS_XA that is available to the public. Therefore, XA clients do not require execute privilege on DBMS_SYSTEM for later oracle versions.
SYS.DBMS_FLASHBACK	Enables self-service repair. If you accidentally delete rows from a table, then you can recover the deleted rows.	For any failure during reconciliation, you can roll back the changes by using this.	This is required for new reconciliation engine in Oracle Identity Manager 11*g* Release 1 (11.1.1) for error handling.
CREATE_MATERIALIZED_VIEW	Creates a materialized view in the grantee's schema	To create the OIM_RECON_CHANGES_BY_RES_MV materialized view	User will not be able to create any materialized view. Only SYS user will be able to do so. This materialized view is required for reporting purpose only.
SELECT ON V$XATRANS SELECT ON PENDING_TRANS$ SELECT ON DBA_2PC_PENDING SELECT ON DBA_PENDING_TRANSACTIONS	Enables an XA Resource Manager and sets privileges so that the XA Resource Manager can manage the interaction between the Oracle database and the applications.	NA	Not recommended to remove. Required for XA support.
ADMINISTER DATABASE TRIGGER	Allows the creation of database-level triggers.	To create DDL trigger named ddl_trigger in Oracle Identity Manager	Users will not be able to create new DDL triggers. It can be removed after schema creation.

23.2 Sample Instance Configuration Parameters

The following sample configuration parameter settings are based on a server with four CPUs (64 bit) and 8 or 20 gigabytes (GB) RAM.

SGA,PGA size are limited by the underlying operating system restrictions on the maximum available memory in some platforms. See Support Note: Oracle Database Server and the Operating System Memory Limitations [ID 269495.1].

Note: In Table 23–2, ASMM denotes the Automatic Shared Memory Management feature available in Oracle Database 10*g* onward. It automatically distributes the memory among various subcomponents to ensure the most effective memory utilization.

You should set the processes parameter to accommodate the following connection pool requirements and few extra connections for external programs:

- Connection pool size of XA data-source configured in Application Server
- Connection pool size for non-XA data-source configured in Application Server
- Direct database connection pool size configured in xlconfig.xml

Table 23–2 Sample Configuration Parameters

Parameter	Recommended Initial Settings for Oracle Database 11*g*
`db_block_size`	8192
`memory_target`	Using Automatic Memory Management feature in Oracle Database 11*g*, the MEMORY_TARGET and MEMORY_MAX_TARGET parameters can be used to manage the SGA and PGA together. Minimum value is 6 GB. For maximum value, use the following formula: MEMORY_TARGET/MEMORY_MAX_TARGET=Total Memory X 80% or 20GB, whichever is greater, assuming that the computer has the database as the primary consumer. When considering MEMORY_TARGET for mangaging the database memory components, SGA_TARGET and PGA_AGGREGATE_TARGET can be left unallocated, which is 0.

Table 23–2 (Cont.) Sample Configuration Parameters

Parameter	Recommended Initial Settings for Oracle Database 11*g*
sga_target	If you use ASMM available in Oracle Database 10g onward, then the SGA components can be managed by specifying the SGA_TARGET and SGA_MAX_SIZE parameters. PGA is managed separately through PGA_AGGREGATE_TARGET. Use any one of the two memory management approaches: ■ MEMORY_TARGET available in Oracle Database 11*g* ■ SGA_TARGET/PGA_AGGREGATE_TARGET available in Oracle Database 10*g* onward Use Oracle ASMM. Minimum value is 4 GB. For maximum value, use the following formula: SGA_TARGET=Total Memory X 80% X 60% or 16 GB assuming an overall memory cap of 20 GB for the Oracle Identity Manager database to run. Assuming that the computer has the Database as the primary consumer. **Note:** These memory paramater values are ballparked figures. As a database administrator, you can also refer to the memory advisors to manage and tune the database.
sga_max_size	10 GB
pga_aggregate_target	Minimum value is 2 GB. For maximum value, use the following formula: PGA_TARGET=Total Memory X 80% X 40% or 4 GB whichever is greater Assuming that the computer has the Database as the primary consumer.
db_keep_cache_size	800M
log_buffer	15 MB
cursor_sharing	FORCE
open_cursors	2000
session_cached_cursors	800
query_rewrite_integrity	TRUSTED
db_file_multiblock_read_count	16
db_writer_processes	2
processes	Based on connection pool settings

23.3 Physical Data Placement

The basic installation of Oracle Identity Manager uses two physical tablespace to store database objects: an (oim_lob) for orchestration-related LOB data and other (oim) for everything else. Oracle Identity Manager database objects belong to one of the following categories:

- Physical tables
- Indexes
- Large objects (LOBs or CLOBs)

Tip: To minimize disk space consumption, Oracle recommends the following:

During the initial startup phase of the deployment, Oracle Identity Manager tablespace is expected to grow at the rate 20G for every 100K users reconciled into Oracle Identity Manager. LOB tablespace grows at around 30% of the size of main Oracle Identity Manager tablespace for the same users. Depending on the usage of orchestration in Oracle Identity Manager, which affects the LOB tablespace growth, the LOB tablespace can grow at a rate of 60% to 100% of the main tablespace in scenarios where orchestration is widely used.

Database administrators must monitor the exact growth rate in the real system for efficient disk space management.

For better performance, create multiple locally managed tablespaces and store each category of database object in a dedicated tablespace. This optimizes storage for efficient data access. Oracle recommends that you place the following User Profile Audit (`UPA`) component tables and indexes in their own tablespaces:

- UPA
- UPA_FIELDS
- UPA_GRP_MEMBERSHIP
- UPA_RESOURCE
- UPA_USR

These tables can store significant amounts of historical data and can be used by historical reports.

The database schema includes the following tables for reconciliation data:

- RCB
- RCE
- RCH
- RCM
- RCP
- RCU
- RPC
- OSI
- SCH
- OSH
- ORC
- OBI
- OUI
- OIO

If your environment generates a large amount of reconciliation data, move these tables to a new tablespace. Use the locally managed tablespaces with automatic extent allocation.

You can use performance metrics to identify tables that are accessed frequently (*hot* tables). To reduce I/O contention, move hot tables to dedicated tablespaces. See "Database Performance Monitoring" on page 23-10 for more information about performance metrics.

Redo-Log Files

Depending on the reconciliation processes configured in Oracle Identity Manager, the volume of database transactions and commits during a reconciliation run can be high. Oracle recommends that you use multiple redo-log files. The total allocated redo-log space should be 1 GB to 2 GB.

Keep Pool Changes

By default, Oracle Identity Manager assigns USR and PCQ tables to be cached in the database by using a keep pool buffer (see db_keep_cache_size in Table 23–2). If your installation contains more than 50,000 users, then Oracle recommends that you use the default database buffer for USR and PCQ tables instead of the keep pool buffer. You can use the following commands.

ALTER TABLE USR STORAGE(buffer_pool default);

ALTER TABLE PCQ STORAGE(buffer_pool default);

23.4 Database Performance Monitoring

To identify performance bottlenecks, you can monitor real-time performance metrics for the Oracle Identity Manager database.

Perform the following at regular intervals:

- Monitor real-time performance by using a performance-monitoring tool such as Oracle Enterprise Manager console or Automatic Workload Repository (AWR) in Oracle Database 11*g*.

  Note: You can use Oracle Enterprise Manager 11g Fusion Middleware Control to monitor Oracle Identity Manager. To do so:

  1. Under Identity Management, select **Oracle Identity Manager** to go to the home page. On the Home page, you can monitor Oracle Identity Manager.
  2. From the Oracle Identity Manager menu, select **Performance** to view performance metrics.

- Collect routine statistics and report by using Oracle Database Enterprise Manager (EM), which is available in Oracle Database 11g (as a standard offering).
  - Routine Stats Gathering

    Routine statistics gathering can be taken care by the 'Automated Maintenance Tasks', which is available in the following navigation path in Oracle Database 11g:

    Oracle EM, the **Server** tab, **Query Optimizer**, **Manage Optimizer Statistics**, the **Automated Maintenance Tasks** link
  - Reporting requirements of stats through Oracle Database 11g EM

    To report on the state of the currently gathered statistics, EM provides a reporting interface in the following navigation path:

Oracle EM, the **Server** tab, **Query Optimizer**, **Manage Optimizer Statistics**, the **Object Statistics** link

This interface can be used for the reporting purpose for All Objects (of the Schema or even the Object of choice), which have Stale, Missing, or Locked states or are already analyzed.

- Collect complete schema statistics upon implementation of Oracle Identity Manager.

  Update schema statistics regularly, so that the Cost-Based Optimizer (CBO) can access the latest statistics. You must consider complete schema or table statistics on mass data change events such as bulkload of users or accounts, import of a new connector, a huge reconciliation run from a new target system, or use of an archival utility.

  This helps the CBO determine an efficient query execution plan that is based on the current state of data. The following is a sample SQL command to collect database statistics on a regular basis:

  See Also: Gathering routine statistics and reporting can be done by performing the automated maintenance tasks available in Oracle Database 11g. See *Oracle Database Performance Tuning Guide 11g Release 1 (11.1)* for details.

```
DBMS_STATS.GATHER_SCHEMA_STATS(OWNNAME=> schema_owner,
  ESTIMATE_PERCENT=>DBMS_STATS.AUTO_SAMPLE_SIZE,
  DEGREE=>8,
  OPTIONS=>'GATHER AUTO',
  NO_INVALIDATE=>FALSE);
```

- Look for relevant recommendations provided in advisory sections in the Automatic Database Diagnostic Monitor (ADDM) or Automatic Workload Repository (AWR) report, and adjust the instance configuration parameters according to the recommended settings. This is specially required after importing a new connector and completing a round of reconciliation from a new target system so that you can identify the need of any new indexes according to your matching rules.

24

Tuning Application Server Performance

This chapter describes how to tune Oracle WebLogic Server for Oracle Identity Manager to improve performance in the following sections:

Note:

- All tuning parameter suggestions and values in this section are for reference purposes only. Values should be modified based on your requirement, application usage patterns, loads, and hardware specifications.
- Changing any of the settings may require you to restart the server.

- JVM Memory Settings
- JDBC Connection Pool
- Number of Message Driven Beans
- User Interface Threads
- Disable Reloading of Adapters and Plug-in Configuration
- Changing the Number of Open File Descriptors for UNIX (Optional)

See Also: Oracle® WebLogic Server Performance and Tuning documentation for more information about tuning Oracle Application Server

24.1 JVM Memory Settings

To change the JVM memory setting:

1. Open the *DOMAIN_HOME*/bin/setSOADomainEnv.sh or setSOADomainEnv.cmd file.
2. Change the value of DEFAULT_MEM_ARGS and PORT_MEM_ARGS from the default value.
3. Save the setSOADomainEnv.sh or setSOADomainEnv.cmd file.

Note: Add the following option to prevent StringIndexOutOfBoundsException error:

```
-XX:-UseSSE42Intrinsics
```

This parameter is required only for Sun JDK.

24.2 JDBC Connection Pool

Oracle Identity Manager uses the oimOperationsDB and oimJMSStoreDS datasources deployed on Oracle WebLogic Server. By default, maximum connections is set at 50. You may have to increase this based on the requirement. To increase the capacity of the JDBC connection pools:

1. Open the WebLogic Server Administration Console.
2. For JDBC Datasource xlXADS:
   a. Click **Services**, **JDBC, Data Sources**, **oimOperationsDB**, and then click the **Connection Pool** tab.
   b. Adjust the Initial Capacity and Maximum Capacity based on requirement.

   For JDBC Datasource xlDS:
   a. Click **Services**, **JDBC**, **Data Sources**, **oimJMSStoreDS**, and then click the **Connection Pool** tab.
   b. Adjust the Initial Capacity and Maximum Capacity based on requirement.
3. Save and activate the changes.

> **Note:** Ensure that any increase in number of connections on the application server connection pools are compensated by database configuration changes. You might have to increase the MAX SESSIONS settings on Oracle Database.

24.3 Number of Message Driven Beans

Oracle Identity Manager uses Message Driven Beans (MDBs) for processing all offline activities, such as reconciliation, auditing, requests, attestation, and for its internal kernel operations. By default, total of 80 MDB instances concurrently serve requests. However, based on the requirement, this can be increased by modifying the OIMMDBWorkManager configuration. To do so:

1. Login to WebLogic Administrative Console.
2. Navigate to **Environment**, **Work Managers**, and then to **MaxThreadsConstraint-1**.
3. Change the count from 80 to a higher number per your requirement.

24.4 User Interface Threads

By default, Oracle Identity Manager provides 20 front-end thread configurations. These threads are used for serving front-end requests. To change the number of front-end thread configurations:

1. Login to WebLogic Administrative Console.
2. Navigate to **Environment**, **Work Managers**, and then to **MaxThreadsConstraint-0**.
3. Change the value of the count from 20 to number per your requirement.

24.5 Disable Reloading of Adapters and Plug-in Configuration

By default, reloading of adapters and plug-in configuration are enabled for ease of development. These should be disabled in the production environment. To do so:

1. Export the /db/oim-config.xml file from MDS as described in "Exporting and Importing Configuration Files" on page 18-1.
2. In the oim-config.xml file, replace the following:

   ```
   <ADPClassLoaderConfig adapterReloadingEnabled="true" loadingStyle="ParentFirst"
   reloadInterval="15" reloadingEnabled="true">
   ```

   With:

   ```
   <ADPClassLoaderConfig adapterReloadingEnabled="false"
   loadingStyle="ParentFirst" reloadInterval="15" reloadingEnabled="false">
   ```

3. Replace the following:

   ```
   <storeConfig reloadingEnabled="true" reloadingInterval="20"/>
   ```

   With:

   ```
   <storeConfig reloadingEnabled="false" reloadingInterval="20"/>
   ```

4. Save the oim-config.xml file and import it back to MDS.

24.6 Changing the Number of Open File Descriptors for UNIX (Optional)

WebLogic limits the number of open file descriptors in the *WEBLOGIC_HOME*/common/bin/commEnv.sh script to 1024. In some cases, if there is a huge number of concurrent users, WebLogic may throw the "TOO MANY OPEN FILES" exception. If you face this error, then increase the limit beyond 1024 in the script. Ensure that the operating system is able to handle the increase in the number of open files.

25

Tuning Connector Performance

This chapter describes how to improve performance by identifying indexes that are required for connector tables and reconciliation tables. It contains the following sections:

25.1 Indexes for Connector Tables

When a connector is imported in Oracle Identity Manager, it creates certain database tables (UD_*) and updates metadata in the Oracle Identity Manager schema. The connector may be further customized to suit processes required in a particular installation with reconciliation rules, data flow, and lookup definitions. After a connector is imported and customized, indexes must be created. The following procedure describes how to identify tables and index key fields. Additional requirements can be gathered by running a reconciliation and examining database AWR reports.

To identify connector tables and index requirements:

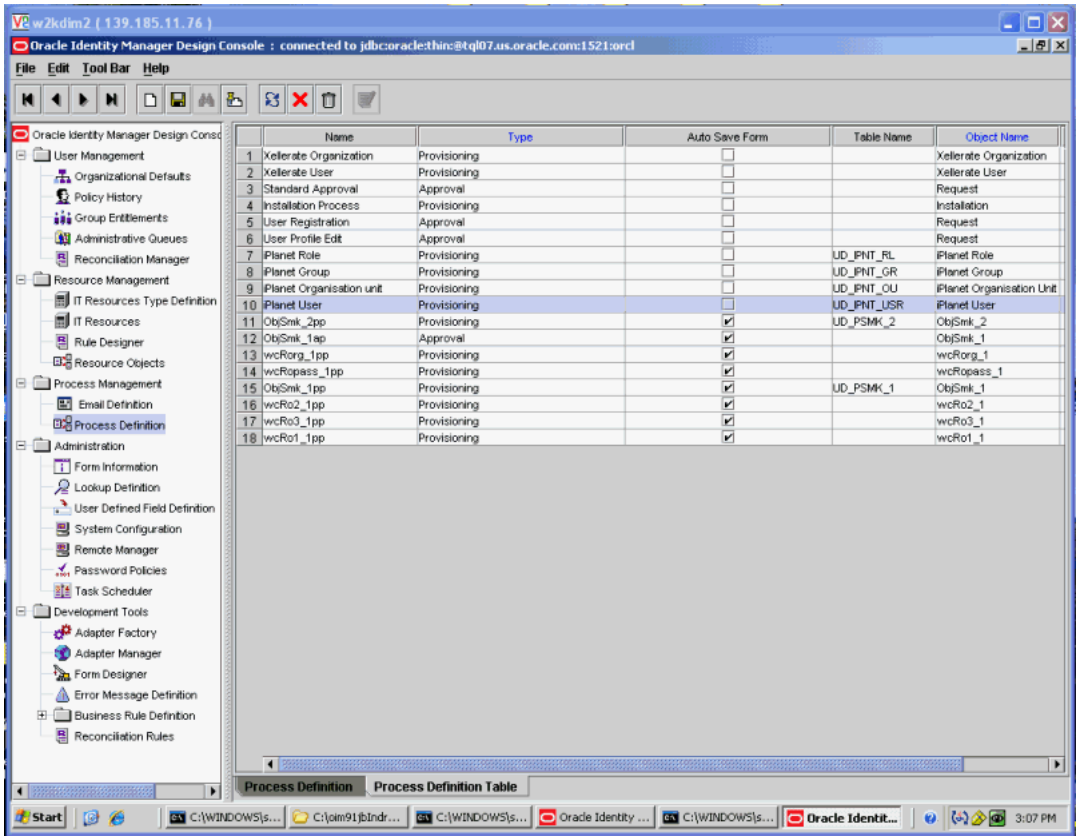
Note: In the following procedure, the Sun Java System Directory connector has been used as an example.

All the key fields used for field mappings must be indexed from the UD_* table or the process definition table.

---

1. Figure 25–1 shows the process definition table for the Sun Java System Directory connector in Oracle Identity Manager Design Console. For this connector, double-click the **iPlanet User** provisioning process, and then click the **Reconciliation Field Mappings** tab to view the field mappings.

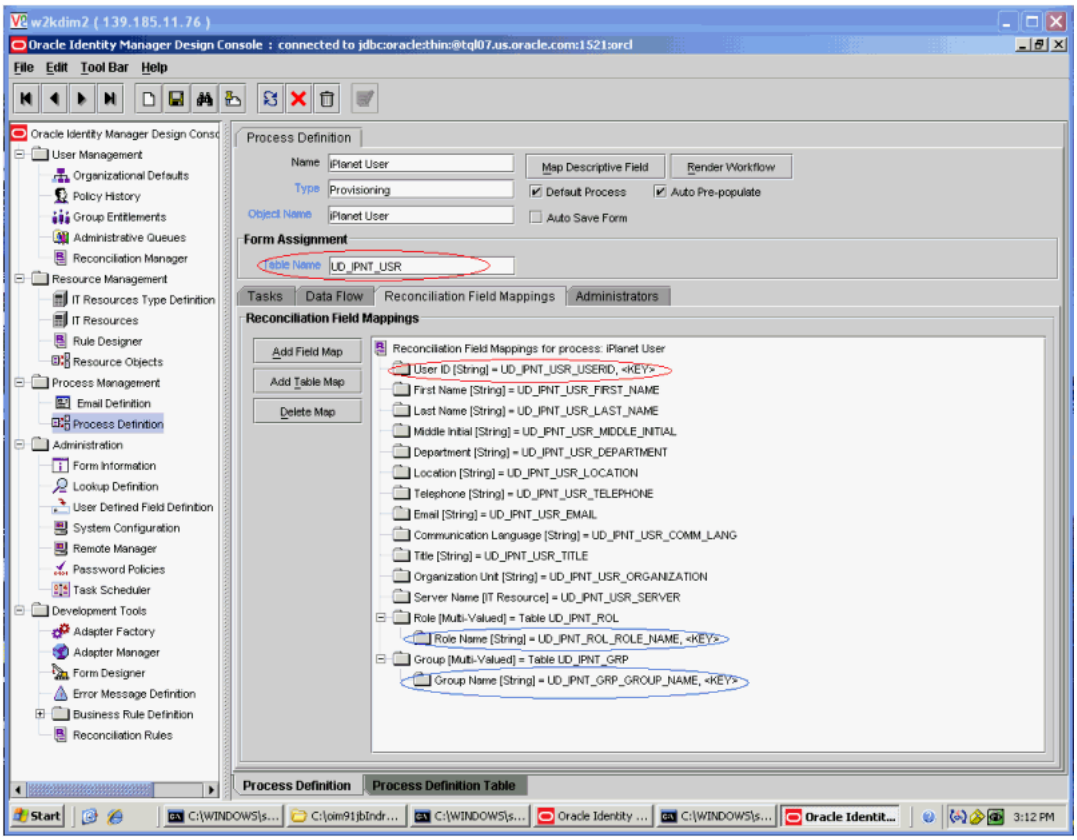
Figure 25–1 Key Fields of a Process Definition Table



2. Figure 25–2 shows the reconciliation field mappings for the Sun Java System Directory connector. In this figure, the table name and the key field are highlighted in red. For this connector, the **UD_IPNT_USR_USERID** column must be indexed.

Note: This is a mandatory step during connector deployment.

Figure 25–2 Reconciliation Field Mappings



Note: if multiple (composite) keys are used for looking up a user, then composite indexes should be created.

The following are the guidelines for indexing key fields:

- The key fields from the child tables must also be indexed. In Figure 25–2, the key fields for child tables are highlighted in blue. For the Sun Java System Directory connector, the UD_IPNT_ROL_ROLE_NAME and UD_IPNT_GRP_GROUP_NAME columns should be indexed.
- If the connector contains any user-defined field and the attribute value is used for searching users in the Oracle Identity Manager database, then the corresponding database field should be indexed.
- If any key field is defined in Oracle Identity Manager as case insensitive, then a function-based index on that key field should be created. For example, if the connector code internally performs a search for the first name (assuming that FIRST_NAME is a key), then the indexing should be performed as follows:

  ```
  CREATE INDEX FDX_USR_FIRST_NAME ON USR(UPPER(FIRST_NAME))
  ```

- While creating indexes, consider using the COMPUTE STATISTICS clause, so that statistics are generated for the index.
- After configuring a connector and creating indexes with above process, you should generate database table and index statistics (or schema statistics).

25.2 Indexes for Reconciliation Tables

For performance tuning, you must create the indexes for the reconciliation tables, as shown in Table 25–1:

Table 25–1 Indexes on Reconciliation Tables

Type of Reconciliation	Table	Index
Identity reconciliation	Target horizontal table	Owner matching Rule column
		RE_KEY
	USR	Owner matching rule column
	RECON_EVENTS	Composite index on RB_KEY and RE_STATUS
	RECON_USR_MATCH	RE_KEY
Account reconciliation	Parent target horizontal table	Account matching rule column
		Entity matching rule column
		RE_KEY
	Child target horizontal table	Child matching rule column
		RE_KEY
	USR	Entity matching rule column
	RECON_EVENTS	Composite index on RB_KEY and RE_STATUS
	RECON_ACCOUNT_MATCH	RE_KEY
	RECON_CHILD_MATCH	RE_KEY

To collect the database statistics:

1. Login to SQL*PLUS as SYS user.
2. Run the following command for the first time:

```
SQL> exec DBMS_STATS.GATHER_SCHEMA_STATS(OWNNAME=>'oimadmin',
ESTIMATE_PERCENT=>DBMS_STATS.AUTO_SAMPLE_SIZE, DEGREE=>8,
OPTIONS=>'GATHER AUTO',NO_INVALIDATE=>FALSE);
```

Note: If the matching rule uses the UPPER clause, then create a functional index on the column. For example, for trusted source reconciliation from LDAP store:

For user reconciliation from LDAP store:

- On the RA_LDAPUSER table, create an index on RECON_ORCLGUID and RE_KEY.
- On the USR table, create an index on USR_LDAP_GUID

26

Tuning and Managing Application Cache

This chapter explains about caching and how it can be managed. It contains the following sections:

26.1 Introduction to Caching

Oracle Identity Manager allows caching of metadata, which reduces DB activities. This results in reduced network load and improved performance.

By default, caching for most of the configurations are disabled (set to false) so that the configuration changes are reflected immediately without having to restart the application servers in the development environments.

26.2 Tuning Oracle Identity Manager Cache

Caching is configured in the /db/oim-config.xml configuration file, which is located in MDS. See Chapter 18, "Using Enterprise Manager for Managing Oracle Identity Manager Configuration" for information about how to make changes to this file.

Oracle recommends the following settings for the production environments for optimal and better performance.

- Set the caching to true for all the components except the following two sections:

  threadLocalCacheEnabled="false"

  "StoredProcAPI" enabled="false"

- Set clustered="false" for non-clustered installation and clustered="true" for clustered installation.

Example 26–1 shows a snippet from the /db/oim-config.xml file, with all the caching enabled for production systems.

Example 26–1 Recommended Cache Values for oim-config.xml in a Clustered Production Environment

```
<cacheConfig clustered="true" enabled="true" expirationTime="144000"
provider="oracle.iam.platform.utils.cache.OSCacheProvider"
threadLocalCacheEnabled="false">
<cacheCategoriesConfig>
<cacheCategoryConfig name="DataObjectEventHandlers" enabled="true"
```

```
expirationTime="14400"/>
<cacheCategoryConfig name="ProcessDefinition" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="EmailDefinition" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="RuleDefinition" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="FormDefinition" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="ColumnMap" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="UserDefinedColumns" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="ObjectDefinition" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="StoredProcAPI" enabled="false" expirationTime="600"/>
<cacheCategoryConfig name="NoNeedToFlush" enabled="true" expirationTime="-1"/>
<cacheCategoryConfig name="MetaData" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="User" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="AdapterInformation" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="OrgnizationName" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="Reconciliation" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="SystemProperties" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="LookupDefinition" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="UserGroups" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="LookupValues" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="ITResourceKey" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="RecordExists" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="ServerProperties" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="ColumnMetaData" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="API" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="CustomResourceBundle" enabled="true"
expirationTime="-1"/>
<cacheCategoryConfig name="CustomDefaultBundle" enabled="true"
expirationTime="-1"/>
<cacheCategoryConfig name="ConnectorResourceBundle" enabled="true"
expirationTime="-1"/>
<cacheCategoryConfig name="LinguisticSort" enabled="true" expirationTime="-1"/>
<cacheCategoryConfig name="GenericConnector" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="GenericConnectorProviders" enabled="true"
expirationTime="-1"/>
<cacheCategoryConfig name="AccessPolicyDefinition" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="UserConfig" enabled="true" expirationTime="-1"/>
<cacheCategoryConfig name="OESDefinition" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="RoleContainerToDescrMap" enabled="true"
expirationTime="-1"/>
<cacheCategoryConfig name="PluginFramework" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="CallbackConfiguration" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="SchedulerTaskDefinition" enabled="true"
expirationTime="14400"/>
<cacheCategoryConfig name="UserStatus" enabled="true" expirationTime="14400"/>
<cacheCategoryConfig name="LocaleCodeLanguageMapping" enabled="true"
expirationTime="14400"/>
</cacheCategoriesConfig>
```

26.3 Purging the Cache

If you want to purge the cache, use the PurgeCache utility in the *OIM_HOME*/bin directory. This utility purges all elements in the cache.

> **Note:** Purging is required when caching is enabled and if you make any system configuration changes. It is not required if caching is disabled.

To use the PurgeCache utility, run PurgeCache.bat <category_name> on Microsoft Windows systems or PurgeCache.sh <category_name> on Linux system. The category name argument represents the name of the category that must be purged. For example, the following commands purge all FormDefinition entries from a system and its clusters:

```
PurgeCache.bat FormDefinition
PurgeCache.sh FormDefinition
```

To purge all Oracle Identity Manager categories, pass a value of "All" to the PurgeCache utility. It is recommended to clear all the categories.

> **Note:** The wlfullclient.jar file must be in the classpath for the PurgeCache utility to run correctly.

Index

A

B

C

D

E

F

H

I

J

L

M

N

O

P

R

S

T

U

V

W