Oracle® Fusion Middleware

Administering Oracle Identity Manager

11*g* Release 2 (11.1.2.3.0)

E56651-11

July 2018

ORACLE®

Oracle Fusion Middleware Administering Oracle Identity Manager, 11*g* Release 2 (11.1.2.3.0)

E56651-11

Primary Author: Anju Poovaiah

Contributing Author: Debapriya Datta

Contributor: Sanjay Rallapalli

Contents

3 Oracle Identity System Administration Interface

Part II Policy Administration

4 Managing Workflows

10 Managing Application Instances

11 Managing Connector Lifecycle

12 Managing Reconciliation

Part VI Requests

13 Managing the Access Request Catalog

14 Managing Home Organization Policy

15 Managing Self Service Capability Policy

16 Managing Lookups

17 Managing Role Categories

18 Managing the Scheduler

19 Managing Notification Service

20 Configuring Oracle Identity Manager

21 Moving From Test to Production

Part VIII Auditing and Reporting

22 Configuring Auditing

23 Using Reporting Features

24 Using the Archival and Purge Utilities for Controlling Data Growth

Part IX Lifecycle Management

25 Handling Lifecycle Management Changes

26 Securing a Deployment

Part X Diagnostics and Troubleshooting

27 Using Enterprise Manager for Managing Oracle Identity Manager

Part XI Appendixes

A Default User Accounts

B Configuring SSO Providers for Oracle Identity Manager

List of Figures

List of Tables

Preface

Oracle Fusion Middleware Administering 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, application servers, connectors, and scheduled task management, connector installation and deployment, and archival utility management.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, refer to the following documents:

- *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*
- *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager*
- *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*
- *Oracle Fusion Middleware Enterprise Deployment Guide 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*

Conventions

The following text conventions are used in this document:

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

What's New In This Guide

This section summarizes the new features and significant changes in *Administering Oracle Identity Manager* in the Oracle Fusion Middleware 11*g* Release 2 (11.1.2.3.0).

Follow the pointers into this guide to get more information about the features and how to use them. This document is the new edition of the formerly titled *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Manager*.

Updates in July 2018 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

Updates in April 2018 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

Updates in January 2018 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* includes the following change:

- Oracle Identity Manager provides TLSv1.2 protocol support for SSL communication between Connector Server and Oracle Identity Manager.

  For changes in the configuration, see Section 25.3, "Configuring SSL for Oracle Identity Manager".

Updates in December 2017 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

Updates in October 2016 Documentation Refresh for 11g Release 2

(11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

Updates in July 2016 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

Updates in April 2016 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

Updates in January 2016 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

Updates in August 2015 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

Updates in June 2015 Documentation Refresh for 11g Release 2 (11.1.2.3.0)

This revision of *Administering Oracle Identity Manager* contains bug fixes and editorial corrections.

New and Changed Features for 11g Release 2 (11.1.2.3.0)

Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) includes the following new and changed administrative features for this document.

- Approval workflow policies, which determine the workflow to be invoked for specific operations. See Chapter 4, "Managing Workflows".
- Disabling SOA workflows and the functional impact of doing so. See Section 4.5, "Running Oracle Identity Manager Without Workflows".
- Home organization policies, which determine the home organizations of the self registering users. See Chapter 14, "Managing Home Organization Policy".
- Self service capability policies, which determine what operations a user can perform for self. See Chapter 15, "Managing Self Service Capability Policy".
- Audit is supported by Oracle Identity Manager for new entities. See Chapter 22, "Configuring Auditing".

- The Orchestration Engine MBean, which helps diagnose and manage various orchestration operations. See Chapter 27, "Using Enterprise Manager for Managing Oracle Identity Manager".

Other Significant Changes in this Document for 11*g* Release 2 (11.1.2.3.0)

For 11*g* Release 2 (11.1.2.3.0), this guide has been updated in several ways. Following are the sections that have been added or changed.

- Revised procedures for creating and managing custom attributes, as a result of the new UI in Oracle Identity Self Service. See Chapter 7, "Configuring Custom Attributes".
- Added information about creating and managing Generic Technology Connectors (GTC). See Chapter 9, "Managing Generic Connectors".
- Added information about creating and managing role categories. See Chapter 17, "Managing Role Categories".
- Added information about configuring various Oracle Identity Manager components, and thereby, control the functionalities of Oracle Identity Manager. See Chapter 20, "Configuring Oracle Identity Manager".
- Added information about moving Oracle Identity Manager deployment from test to production. See Chapter 21, "Moving From Test to Production".
- Added information about some SSO integration use cases. See Appendix B, "Configuring SSO Providers for Oracle Identity Manager".
- Moved information about managing password policies to *Performing Self Service Tasks with Oracle Identity Manager*. See "Managing Password Policy" section in that book.
- Removed information about managing attestation processes as attestation is not supported in this release.
- Moved information about managing identity certification to *Performing Self Service Tasks with Oracle Identity Manager*. See "Managing Identity Certification" section in that book.
- Removed information about installing and configuring a remote manager as using a remote manager is not recommended.
- Removed information about using the Form Upgrade Job and Form Version Control Utility. Although fresh Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) deployment may not need the Form Upgrade Job to be run, it can still be run for upgrading older versions of forms to the latest version in this release of Oracle Identity Manager depending upon whether you have upgraded from prior Oracle Identity Manager releases where older versions of the same form were used. For information about the Form Upgrade Job utility, refer to the following URL:

  `http://docs.oracle.com/cd/E40329_01/admin.1112/e27149/formver.htm#OMADM2163`

Part I

Overview

This part describes the Oracle Identity Manager overview and architecture and provides an overview of the Oracle Identity System Administration interface.

It contains the following chapters:

- Chapter 1, "Product Overview"
- Chapter 2, "Product Architecture"
- Chapter 3, "Oracle Identity System Administration Interface"

1 Product Overview

This chapter describes the purpose of Oracle Identity Manager and highlights the major features. The chapter includes the following topics:

- What is Oracle Identity Manager?
- What are the Different Modes of Oracle Identity Manager?
- How does Oracle Identity Manager Interact with Other IT Systems?
- How does Oracle Identity Manager Interact with Other Oracle Identity and Access Management Products?
- How do Users Interact with Oracle Identity Manager?

1.1 What is Oracle Identity Manager?

Oracle Identity Manager is a Governance solution that provides self service, compliance, provisioning and password management services for applications residing on-premise or on the Cloud.

Oracle Identity Manager makes it possible for enterprises to manage the identities and access privileges of their customers, business partners, and employees, all on a single platform. It allows these users to manage their own identities as well as those of others by using delegated administration. It allows enterprises to setup delegated administrators, who are users empowered to manage the identities, passwords, password policies, and access of other users. Business users can create and manage the lifecycle of enterprise roles, which grant access to end-users. These roles can be granted automatically by using rules. With the help of roles and access policies, organizations can ensure that their users are on-boarded and off-boarded in a timely and automated manner.

Oracle Identity Manager enables end-users to get the access they need to do their jobs in a simple and user-friendly manner. End-users use the access catalog, which presents available access in a non-technical, user-friendly manner, to request the access they need. They submit their requests, which are routed to approvers and managers for approval.

Oracle Identity Manager automates the process of creating, updating, and deleting user accounts, provisioning of passwords, and granting/revoking of entitlements across applications hosted on the Cloud or on-premise. This process is known as provisioning and de-provisioning. Oracle Identity Manager makes use of connectors to do provisioning and de-provisioning with connected applications. It also supports manual provisioning and de-provisioning in applications that do not support a connector. Such applications are called disconnected applications.

Oracle Identity Manager can synchronize identities from authoritative sources, such as HR applications and accounts and access privileges from applications including LDAP and databases. Identity lifecycle events, such as hire, transfer, manager change, and separation from the organization, can be synchronized with Oracle Identity Manager, which can then take appropriate action including revoking access. This mechanism of synchronizing identity information with an authoritative source of identity data is known as trusted reconciliation. Oracle Identity Manager can also synchronize account information, including access privileges, and entitlements from applications that it manages. This mechanism is known as target reconciliation.

Oracle Identity Manager helps managers, authorized users, and compliance administrators to review and certify user access, in a user-friendly manner, by a process known as identity certification. Authorized administrators can create and configure certification campaigns, on a scheduled or ad-hoc basis, by using simple wizards. Certifiers, who have to certify the user access, are presented the information in a simple manner. They can either approve the access or reject it. When a violation is detected and the access is rejected, Oracle Identity Manager initiates a process that enables administrators to correct the violation. It can also directly deprovision the access privileges from the target platform or application, while maintaining a comprehensive trail of the actions taken. This is known as closed-loop remediation. Oracle Identity Manager supports different types of certifications, based on various user personas, such as business managers, role owners, application owners, and entitlement owners.

Oracle Identity Manager makes it possible for organizations to meet their compliance objectives by allowing business users to define audit policies. Audit policies specify what type of access a user may or may not have. For example, a user who has access to both Accounts Payables and Accounts Receivables is violating Sarbanes-Oxley guidelines. This is known as a Segregation of Duties (SoD) violation. Oracle Identity Manager allows organizations to define SoD policies that can be enforced during access request and can also be used to scan existing access to identify toxic combinations of access privileges, known as policy violations. Oracle Identity Manager identifies the violations and initiates a workflow allowing remediators, who could be business manager or administrators to fix these violations. This process is known as remediation. All actions taken by remediators are recorded and a comprehensive audit trail is maintained.

Oracle Identity Manager provides comprehensive auditing capabilities that allow auditors and security staff to keep track of who initiated what change, on whom, when and in what context. It allows the creation of custom audit events. This enables customers to audit their workflows and processes. All audit information is available in a manner that can be reported on using standard reporting tools. Oracle Identity Manager provides an embedded reporting server, which delivers print-quality reports for most product areas including request and approvals, password management, identity certification, and identity audit. Customers have the flexibility of using their own enterprise reporting tool as well.

1.2 What are the Different Modes of Oracle Identity Manager?

Oracle Identity Manager provides the flexibility to use functionality based on your identity management requirements. You can enable specific functionality by picking specific deployment options. Oracle Identity Manager can be configured in three deployment modes:

- **Oracle Identity Manager in database mode**

Oracle Identity Manager is a highly scalable identity administration and provisioning solution that is capable of managing millions of identities, roles, and entitlements, and thousands of applications that are stored in a database. This mode should be used when identity administration, access request, account, and entitlement provisioning and reconciliation is the main business driver and simple Single Sign On (SSO) with a SSO solution is adequate.

- **Oracle Identity Manager with Identity Auditor mode enabled**

  Oracle Identity Manager with the Identity Auditor mode enabled provides the ability to run certification campaigns, manage and make use of identity audit policies, and carry out role mining to detect clusters of roles and policies.

  Identity Auditor mode enables you to use the role LCM, Segregation of Duties (Identity Audit), and Access Certification features. You must be licensed to use the Identity Auditor features.

  Note: Identity Auditor mode can be enabled after installing Oracle Identity Manager. See "Enabling Identity Audit" in *Performing Self Service Tasks with Oracle Identity Manager* for information about enabling the Identity Auditor mode.

Table 1–1 provides a summary of the features that are available in each deployment mode of Oracle Identity Manager.

Table 1–1 Summary of Features

Feature	Oracle Identity Manager in DB mode	Oracle Identity Manager with Identity Auditor mode enabled
Access policy management	Yes	Yes
Access request	Yes	Yes
Approvals	Yes	Yes
Auditing	Yes	Yes
Delegated administration	Yes	Yes
Identity audit (SoD)	No	Yes
Identity certification	No	Yes
Identity store	Database	Database
Lost password, forgot user ID, self registration	Yes	Yes
OAM/OAAM/OMSS integration	Yes	Yes
Organization management	Yes	Yes
Password synchronization	Yes	Yes
Provisioning	Yes	Yes
Reconciliation	Yes	Yes
Reporting	Yes	Yes
Role management	Yes	Yes
User management	Yes	Yes

Table 1–1 (Cont.) Summary of Features

Feature	Oracle Identity Manager in DB mode	Oracle Identity Manager with Identity Auditor mode enabled
User password management	Yes	Yes

Note:

- Workflows can be disabled in all modes. However, certain features require workflows. See "Running Oracle Identity Manager Without Workflows" on page 4-31 for information about disabling workflows and the impact of doing so on various Oracle Identity Manager features.
- See "Configuring Auditing" on page 22-1 for information about auditing.

1.3 How does Oracle Identity Manager Interact with Other IT Systems?

In Oracle Identity Manager, applications and other IT systems are called *IT resources*. The IT resources expose various objects that can be managed by Oracle Identity Manager. These objects are called *resource objects*. The objects that represent accounts are called *application instances*, and the objects that represent access within an application are known as *entitlements*.

Oracle Identity Manager interacts with various applications and IT systems to manage the application instances and accounts by using connectors. Connectors are installed on the Oracle Identity Manager Server. Oracle provides several connectors for common technologies, such as JDBC, LDAP, SPML, SOAP, and REST, and for common business applications, such as SAP, eBusiness Suite, and PeopleSoft. New connectors can be developed by using the Identity Connector Framework (ICF).

Some IT systems cannot be communicated with directly and require the use of a lightweight component called the Connector Server. Examples of applications that require the use of the Connector Server include Microsoft products, such as Exchange and Active Directory, Novell eDirectory, IBM Lotus Notes, and others. In such scenarios, the connector is deployed on the Connector Server, and it communicates using native protocols with the application. Oracle Identity Manager communicates with the Connector Server, which then communicates with the connector.

1.4 How does Oracle Identity Manager Interact with Other Oracle Identity and Access Management Products?

Oracle Identity Manager integrates with other Oracle and third-party Identity and Access Management products via standards-based integration. However, Oracle Identity Manager provides a default integration with the following products in the Oracle Identity and Access Management Suite:

- **Integration with Oracle Privileged Account Manager (OPAM)**

  Oracle Privileged Account Manager (OPAM) allows organizations to protect highly-privileged application accounts as well as shared access to sensitive applications.

When Oracle Identity Manager is integrated with OPAM, it manages the identity lifecycle of the users who have access to OPAM. It allows users to request access to the privileged accounts in various applications by using the access catalog feature. Administrators in Oracle Identity Manager can control which users can request access to which applications. They can also make it easier for users to request access to privileged accounts by enhancing the business metadata in the access catalog.

Oracle Identity Manager allows organizations to certify the privileged access that users have with the help of Identity Certification. Managers and authorized users can see who has access to what and can take the appropriate decision to certify or reject the access.

- **Integration with Access Manager**

  When integrated with Oracle Access Manager (OAM), Oracle Identity Manager provides forgot user ID, forgot password, challenge questions and responses, password and password policy management, account locking, self registration, and user, role, and organization management services. OAM provides Single Sign On services for Oracle Identity Manager. OAM also provides real-time session kill if the user is locked and auto-unlock features.

  Oracle Identity Manager requires the use of the LDAP synchronization feature. This feature allows Oracle Identity Manager to push users, user passwords, and changes to user attributes, groups, and group memberships to the LDAP directory. Oracle Identity Manager reconciles the changes from the LDAP directory including the account lock status.

  Oracle Identity Manager supports a reduced and simplified integration with OAM as well, where OAM provides Single Sign On for Oracle Identity Manager. In this approach, there is no synchronization of the state attributes or of OIM users and groups. You can make use of provisioning and connectors to provision and reconcile LDAP users and groups.

- **Integration with Access Manager and Adaptive Access Manager**

  When used with OAM and Oracle Adaptive Access Manager (OAAM), you can leverage the Knowledge-based Authentication (KBA) that provides a rich set of challenge questions, configurable logic behind presenting those questions to the user, and validating the responses. Oracle Identity Manager delegates the management of challenge questions and answers to OAAM. This integration also allows users to reset their challenge questions.

- **Integration with Oracle Mobile Security Suite**

  Oracle Mobile Security Suite (OMSS) allows organizations to extend their identity platform and security policies to mobile devices. It provides a secure container for application security and control. It allows corporate applications and data to be isolated from personal applications and data without needing to lock down the entire device.

  When integrated with Oracle Identity Manager, it allows users to see their devices and applications. Administrators can manage devices, applications, and mobile policies. They can also configure the device remotely and applications to specific groups of users.

Note: For details about the integration with various products, see *Oracle Fusion Middleware Integration Guide for Oracle Identity Management Suite*.

1.5 How do Users Interact with Oracle Identity Manager?

Oracle Identity Manager provides an end-user interface, called the Identity Self Service console, and a system administrator interface, called the Identity System Administration console. Both end-users and system administrators use the web browser to log on to Oracle Identity Manager.

The interface for end-users is used:

- To manage your user profile, passwords, challenge questions, and account passwords.
- To view, request, and approve access for self and others, certify users, and process policy violations and manual provisioning tasks.
- To setup organizations and administration roles and to configure delegated administration. It is also used by delegated administrators to create and manage users, organizations, and password policies.
- By authorized users to compose roles, create and run certification campaigns, configure SoD rules and policies, and create and run compliance scans.

The interface for system administrators is used:

- To define workflow policies, home organization policies, and user capabilities
- To manage the schema of system entities, such as user, role, and organization
- To manage provisioning end-points and the schema of the supported objects
- To import/export Oracle Identity Manager configuration objects
- To install/uninstall/upgrade connectors

You can also use the REST services to either create your own user interface or to integrate other applications with Oracle Identity Manager.

Developers can also use:

- The JDeveloper IDE to create custom UI by using the Oracle Application Development Framework (ADF) and to create custom workflows by using Business Process Execution Language (BPEL)
- The Design Console, which is a Java thick client, to create provisioning workflows
- The embedded BI Publisher reporting server to create custom reports

2

Product Architecture

This chapter provides an overview of Oracle Identity Manager product architecture. It consists of the following topics:

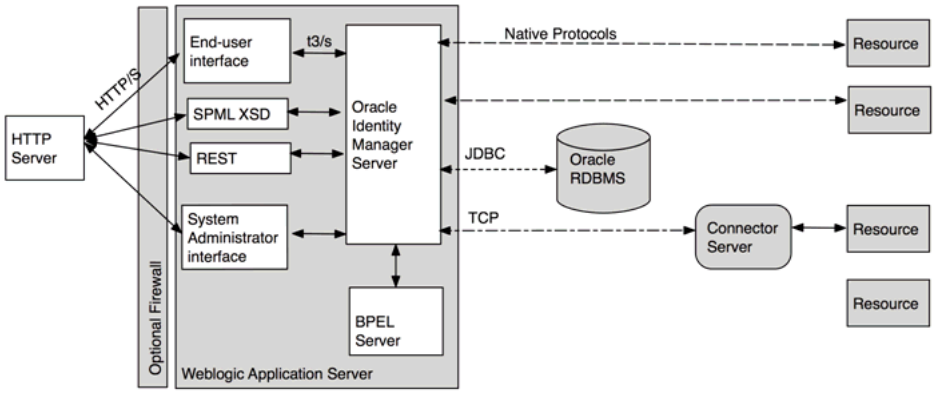
- Oracle Identity Manager Components
- Multi-tiered Architecture

2.1 Oracle Identity Manager Components

Oracle Identity Manager is a J2EE web application. The J2EE platforms consists of a set of industry-standard services, APIs, and protocols that provide the functionality for developing multi-tiered and web-based enterprise applications.

Figure 2–1 shows the various components of Oracle Identity Manager.

Figure 2–1 Oracle Identity Manager Components



2.2 Multi-tiered Architecture

The system architecture of Oracle Identity Manager is distributed across logical tiers, as described in the following sections:

- Understanding the User Interface Tier
- Understanding the Application Tier
- Understanding the Database Tier
- Understanding the Connector Tier

2.2.1 Understanding the User Interface Tier

The user-interface tier (or the user tier) consists of administrators and end-users who interact with Oracle Identity Manager through one of the user interfaces. The main user interface for Oracle Identity Manager is web-based, which communicates with Oracle Identity Manager over HTTP/S. There are two browser-based UIs, the end-user facing Oracle Identity Self Service and the administrator facing Oracle Identity System Administration. These UIs are developed by using the Oracle Application Development Framework (ADF).

Identity Self Service can be customized via the web browser, by system administrators who can add links, add business logic to show/hide form fields, extend shipped forms, and perform several other common UI customization tasks. Administrators perform UI customization tasks in UI sandboxes. These sandboxes can be exported and imported into higher environments. The use of Oracle ADF and UI customization framework allows administrators to customize Identity Self Service in an upgrade-safe manner.

Identity System Administration allows administrators to perform typical system administration functions including scheduling jobs, onboarding applications, and managing schemas. This UI is not customizable.

Developers can use the Design Console to create provisioning workflows and Oracle JDeveloper to create BPEL workflows for manual fulfillment, approval, identity certification, and identity audit.

2.2.2 Understanding the Application Tier

Oracle Identity Manager Server is a J2EE application. It is deployed on Oracle WebLogic Server. The server consists of the Identity Self Service and Identity System Administration web applications, SPML XSD, and REST services, and the EJBs and related Java classes that provide the core functionality. Connectors, which interact with other IT systems, are deployed on the Oracle Identity Manager Server.

Note: Oracle recommends that you use REST services instead of SPML.

The server comprises of the following functional components:

- **Identity administration**

  This includes self-registration, lost password and forgotten user ID, user, role, and organization management, and password management.

  The user management engine allows administrators to manage users; reset their passwords and grant/revoke/modify access. When integrated with Oracle Access Manager (OAM), the changes in the user profile are synchronized with the LDAP directory used by OAM using a feature called LDAP synchronization.

  The role management engine allows business users and administrators to create static and dynamic roles, associate access via access policies, and make the role available to various organizations. These operations can go through approval. After approval, the changes are committed to the Oracle Identity Manager repository. This feature is known as role lifecycle management.

  The organization management engine allows administrators to create and manage static or rule-based dynamic organizations. Administrators can define password

policies and associate them with organizations, which allows different user communities to have different password policies.

- **Authorization**

  The authorization engine in Oracle Identity Manager allows granular delegated administration by allowing administrators to define admin roles and associate them with functional capabilities. The authorization engine enforces the policies, which in turn leverage the admin role memberships of the user. Administrators can also define attribute-level permissions for users and specify who can see and modify user attributes.

- **Provisioning and reconciliation**

  Oracle Identity Manager provides a highly scalable provisioning engine that provides account management and account password management capabilities. Oracle Identity Manager allows administrators to manage accounts and grant/revoke/modify additional access (entitlements). Administrators and end-users can also reset account passwords or configure Oracle Identity Manager so that the user password is synchronized with the accounts provisioned to a user. The provisioning engine supports two types of provisioning, connected provisioning using connectors and disconnected provisioning (or manual fulfillment) where a user has to take some action.

  The reconciliation engine allows changes in target applications to be detected and synchronized with Oracle Identity Manager. It can retrieve changes from an authoritative source or from a target resource. In the former scenario, changes are synchronized with the user, while in the latter, with the account.

- **Access request and approvals**

  The request engine allows end-users to submit requests for new and modified access, either for themselves or for others. They can use the access catalog to search and browse in a manner similar to online shopping and submit their requests. The requests are routed to the appropriate approvers and fulfilled either in an automated manner by using connectors, or manually by using disconnected provisioning.

- **Identity certification**

  The identity certification engine allows administrators to define certification campaigns. These campaigns allow managers and authorized users to review and certify the access granted to users. They can delegate certain users or process them themselves. They can reject a user's access, which can trigger a provisioning action to revoke the access. This is called closed-loop remediation.

- **Identity audit or Segregation of Duties (SoD)**

  The SoD engine allows administrators to define rules and group them into policies. These rules and policies, known as identity audit rules and policies, allow Oracle Identity Manager to detect access that violates compliance rules. Administrators can specify which policies should be enforced during access request, while allowing other policies to be enforced retroactively. When a policy violation is found, the engine assigns the violation to a user for remediation.

- **Auditing**

  The auditing engine audits (or logs) various actions in Oracle Identity Manager. Administrators can also add custom audit events. The audit data can be reported on using the reporting capabilities of Oracle Identity Manager.

- **Embedded reporting server**

The embedded reporting server, based on Oracle BI Publisher, provides operational and historical reports. Administrators can also use standalone BI Publisher or use the schema information to create reports using any other reporting tool.

- **BPEL workflow engine**

  Oracle Identity Manager uses BPEL to provide workflow orchestration for approval, manual fulfillment, identity certification, and identity audit. Administrators or developers can define BPEL workflows or SOA composites and use workflow rules to dynamically invoke these workflows. BPEL provides data-driven approver resolution, task expiration, and escalation and email-based actionable notification. Oracle JDeveloper can be used to create new workflows and register them in Oracle Identity Manager.

2.2.3 Understanding the Database Tier

Oracle Identity Manager stores all its information in the Oracle Identity Manager repository. The repository is comprised of tables that store the configuration, state, and other data. Oracle Identity Manager keeps a copy of the account and entitlement data that is provisioned to the user, allowing it to be the source of truth for identity and account data.

Oracle Identity Manager also makes use of other schemas to store metadata about the workflows, approvals, configuration, and authorization policies.

Because Oracle Identity Manager can accumulate state data, it provides archival and purge utilities to manage data growth. Administrators must follow the product recommendations to manage data growth for optimal performance.

2.2.4 Understanding the Connector Tier

The connector tier consists of applications and IT systems to which you provision and deprovision user accounts, change the account password, and grant/revoke entitlements. It includes the connector Server, which is a lightweight application that allows Oracle Identity Manager to manage applications that do not provide remote APIs or require native integration.

Typically, Oracle Identity Manager connectors are developed by using the Identity Connector Framework and are deployed with the server. In some cases, where a connector server is required, they are deployed on the connector server.

You can create your own connectors by using the Identity Connector Framework, a lightweight and easy to use framework for developing connectors.

3

Oracle Identity System Administration Interface

This chapter discusses the procedure to access and log in to Oracle Identity System Administration, and provides an overview of the Oracle Identity System Administration.

This chapter contains the following topics:

- Logging in to Oracle Identity Manager System Administration Console
- Overview of the Oracle Identity Manager System Administration Console

3.1 Logging in to Oracle Identity Manager System Administration Console

To log in to Oracle Identity Manager System Administration Console:

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

   `http://HOSTNAME:PORT/sysadmin`

   In this URL, *HOSTNAME* represents the name of the computer hosting the application server and *PORT* refers to the port on which the server is listening.

   Note: The application name, sysadmin, is case-sensitive.

2. After the Oracle Identity Manager System Administration Console login page is displayed, log in with your user name and password.

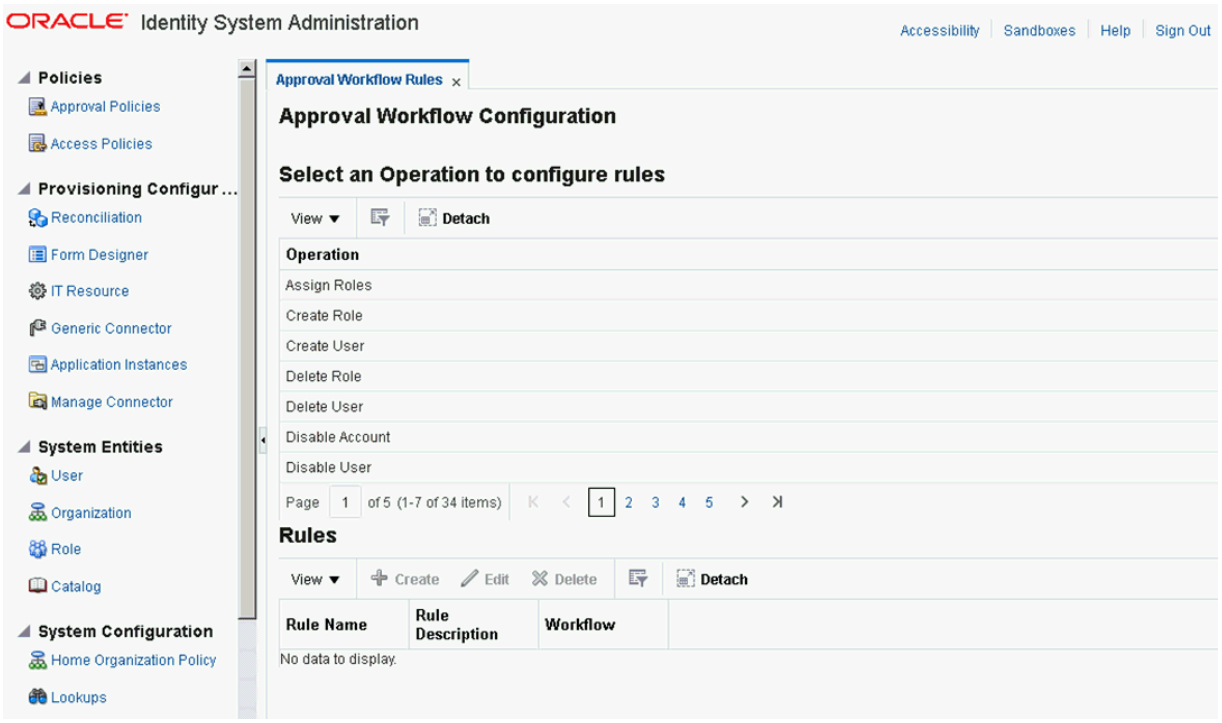
3.2 Overview of the Oracle Identity Manager System Administration Console

The interface of the Oracle Identity System Administration is composed of the following areas:

- Links
- Left and Right Panes

Figure 3–1 shows a sample page and the layout of the interface.

Figure 3–1 Layout of the Oracle Identity System Administration Console



3.2.1 Links

This area consists of the following links in the upper-right corner of the interface:

- Accessibility
- Sandboxes
- Help
- Sign Out

3.2.1.1 Accessibility

The Oracle Identity Manager System Administration Console interface has been designed to adhere to the standards set in Section 508 of the Rehabilitation Act and the World Wide Web Consortium's Web Content Accessibility Guidelines 2.0 AA (WCAG 2.0 'AA").

When you click the Accessibility link in the upper right corner of the page, the Accessibility dialog box is displayed. You can select one of the following options from the Accessibility dialog box:

- **I use a screen reader**

  Select this option if you want to use a screen reader.

- **I use high contrast colors**

  Select this option to use the high-contract color scheme that you have specified in your operating system, rather than using the default color scheme specified in the Oracle Identity Manager System Administration Console.

- **I use large fonts**

Select this option if you want to change the font size for easy viewing and readability.

3.2.1.2 Sandboxes

A sandbox represents an area where metadata objects can be modified without affecting their mainline usage. In other words, a sandbox is a temporary storage area to save a group of runtime page customizations before they are either saved and published to other users, or discarded.

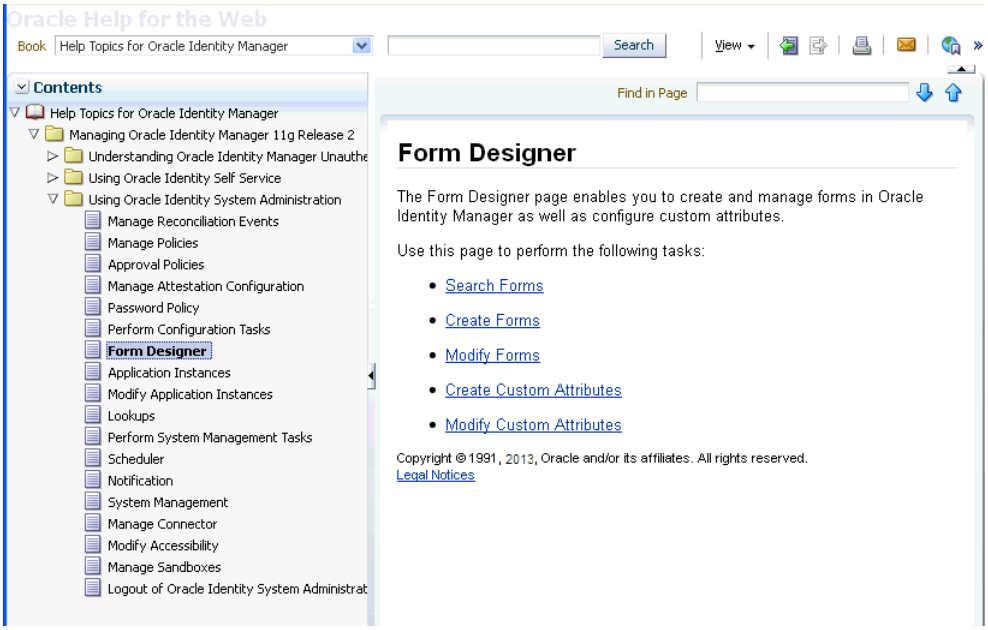
In the Manage Sandboxes page, you can create, delete, activate, deactivate, and publish sandboxes. See the "Managing Sandboxes" section in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for detailed information.

3.2.1.3 Help

The Oracle Identity Manager System Administration Console interface includes a help system. Clicking the Help link opens the help system in a new window. In addition, this interface provides context-sensitive help. For example, if you are in the Form Designer page and click the Help link, then help content related to form designer is displayed.

Figure 3–2 shows a sample page and default layout of the help interface.

Figure 3–2 Layout of the Help Interface



The default view of the help system consists of three panes:

- Top Pane
- Lower Left Pane
- Lower Right Pane

3.2.1.3.1 Top Pane

The top pane consists of the following:

- Book drop-down list: From this drop-down list you can select one of the following values:
  - **Help Topics for Oracle Identity Manager:** Select this value to open all help topics for Oracle Identity Manager.
  - **Administrator's Guide for Oracle Identity Manager:** Select this value to open the online help version of *Oracle Fusion Middleware Administrator's Guide for Oracle Identity Manager*.
  - **Developer's Guide for Oracle Identity Manager:** Select this value to open the online help version of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
  - **User's Guide for Oracle Identity Manager:** Select this value to open the online help version of *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager*.
  - **Custom Help Topics for Oracle Identity Manager:** Select this value to open any custom help topics.
- Search field: Specify any word or term to search for in the help system.
- View: From the View menu, you can select any one of the following options:
  - **Maximize Reading Pane:** Collapses the lower left pane to maximize the reading pane, which is the lower right pane.
  - **Restore Default Window Layout:** Restores the current layout of the help system to the default layout.
  - **Contents:** Restores the lower left pane to display the Contents region along with the help topics, if it is not already being displayed.
  - **Search:** Displays the Search region in the lower left pane. In the Search region, you can search for help topic and the search results are displayed in a tabular format. Here are a few guidelines on performing a search:
    * Search criterion specified in the Search field can be made case sensitive by selecting the **Case Sensitive** option.
    * To define your search precisely, you can specify the boolean operators & (for AND), | (for OR), ! (for NOT) in your search criterion, select the **Boolean expression** option, and then click **Search.**
    * To search for help topics containing all words specified in the search criterion, select **All words.**
    * To search for help topics containing any word specified in the search criterion, select **Any words.**
  - **Show permanent link for this topic page:** If you want to save the link to a help topic for future reference, then from the View menu, select **Show permanent link for this topic page.** In the dialog box that is displayed, right-click the link to the help topic and select one of the following options:
    * **Bookmark This Link:** Adds the help topic URL to the browser bookmarks.
    * **Copy Link Location:** Copies the help topic URL to the clipboard.
- Toolbar: The help system contains a toolbar that provides action buttons for certain tasks. You can view the name of the button by moving the mouse pointer over the button. The following buttons are available:

- **Go back one page:** Takes you back to the page containing the previous help topic.
- **Go forward one page:** This icon is enabled only if you have clicked the **Go back one page** icon. Clicking the **Go forward one page icon** takes you to the next page in the sequence of topics you visited.
- **Print this topic page:** Prints the current help topic.
- **Email this topic page:** Drafts an email with a link to the help topic currently displayed in the help system. This draft can be sent to the desired email recipient.
- **Link to this topic page:** Saves the link to a help topic for future reference by right-clicking the link to the help topic in the dialog box that is displayed, and then selecting one of the following options:
  * **Bookmark This Link:** Adds the help topic URL to the browser bookmarks.
  * **Copy Link Location:** Copies the help topic URL to the clipboard.

3.2.1.3.2 Lower Left Pane

The lower left pane contains the Contents and Search regions. By default, the Contents region is expanded. The Contents region displays links to help topics depending on the option you select from the Book drop-down list in the top pane. You can click the arrow icon beside Contents to expand or collapse the Contents region.

3.2.1.3.3 Lower Right Pane

The lower right pane displays any help topic that you search for or open from the Contents and Search regions in the lower left pane. This pane is also known as the reading pane.

3.2.1.4 Sign Out

Click the Sign Out link to log out of Oracle Identity System Administration.

3.2.2 Left and Right Panes

Every page in the Oracle Identity System Administration is divided into two panes. The left pane consists of sections that contain links to regions using which a variety of tasks can be accomplished. The left pane is the primary navigation tool and is displayed on all web pages of Oracle Identity System Administration. Depending on the link that you click in the left pane, corresponding details are displayed in the right pane.

The left pane consists of these regions:

- Policies
- Provisioning Configuration
- System Entities
- System Configuration
- Upgrade
- Workflows

3.2.2.1 Policies

The Policies region contains the following:

- Approval Policies

  Use this page to create and manage approval policies if you have upgraded Oracle Identity Manager from an earlier release. An approval policy helps to associate request types with approval processes defined in the workflow service.

  See the following URL for more information:

  https://docs.oracle.com/cd/E40329_01/admin.1112/e27149/appr_policies.htm#OMADM2264

- Access Policies

  Use this page to create and manage access policies. Access policies define how to automate the provisioning of target systems to users.

  See "Managing Access Policies" on page 5-1 for more information.

3.2.2.2 Provisioning Configuration

The Provisioning Configuration region contains the following:

- Reconciliation

  Use the Reconciliation page to create and manage reconciliation events. See "Managing Reconciliation Events" on page 12-10 for more information.

- Form Designer

  Use this page to create and manage forms of type users, roles, organizations, catalog, and resources that are not predefined in Oracle Identity Manager.

  See "Managing Forms" on page 6-1 for more information.

- IT Resource

  Use this page to create and manage IT resources. An IT resource is composed of parameters that store connection information about a target system. Oracle Identity Manager uses this information to connect to a specific installation or instance of the target system.

  See "Managing IT Resources" on page 8-1 for more information.

- Generic Connector

  Use the Generic Connector page to create and manage generic connectors. Generic connectors are basic connectors without advanced features. The connectors utilize generic connectivity technologies, such as SPML and JDBC.

  See "Managing Generic Connectors" on page 9-1 for more information.

- Application Instances

  Use this page to create and manage application instances. An application instance is a combination of an IT resource instance and resource object. Users have accounts and entitlements that are associated with application instance and not with the IT resource instance or resource object.

  See "Managing Application Instances" on page 10-1 for more information.

- Manage Connector

  Use this page to define, install, clone, upgrade, and uninstall predefined connectors in an Oracle Identity Manager environment. A predefined connector is

designed for commonly used target systems such as Microsoft Active Directory and PeopleSoft Enterprise Applications.

See "Managing Connector Lifecycle" on page 11-1 for more information.

3.2.2.3 System Entities

The System Entities region contains the following:

- User

  Click to customize the User form, such as to create a UDF for the user entity.

- Organization

  Click to customize the Organization form, such as to create a UDF for the organization entity.

- Role

  Click to customize the Role form, such as to create a UDF for the role entity.

- Catalog

  Click to customize the Catalog form, such as to create a UDF for the catalog entity.

3.2.2.4 System Configuration

The System Configuration region contains the following:

- Home Organization Policy

  Use this page to create and manage policies based on which the home organization of a user is determined at the time of self registration.

  See "Managing Home Organization Policy" on page 14-1 for more information.

- Self Service Capabilities

  Use this page to define self service capability policies to control what operations a user can perform for self.

  See "Managing Self Service Capability Policy" on page 15-1 for more information.

- Lookups

  Use this page to create and manage lookup definitions. See "Managing Lookups" on page 16-1 for more information.

- Role Categories

  Use this page to create and manage role categories for categorizing roles for the purpose of navigation and authorization.

  See "Managing Role Categories" on page 17-1 for more information.

- Scheduler

  Use this page to create and manage scheduled jobs. Scheduled jobs are jobs that are run at specified time intervals to manage various activities in Oracle Identity Manager.

  See "Managing the Scheduler" on page 18-1 for more information.

- Notification

  Use this page to create and manage notification templates. A notification template is used to send notifications.

See "Managing Notification Service" on page 19-1 for more information.

- Configuration Properties

  Use this page to create and manage system properties. System properties define the characteristics that control the behavior of Oracle Identity Manager.

  See "Configuring Oracle Identity Manager" on page 20-1 for more information.

- Import

  Use this page to import Oracle Identity Manager configurations by using the Deployment Manager.

  See "Migrating Incrementally Using the Deployment Manager" on page 21-1 for more information.

- Export

  Use this page to export Oracle Identity Manager configurations by using the Deployment Manager.

  See "Migrating Incrementally Using the Deployment Manager" on page 21-1 for more information.

3.2.2.5 Upgrade

When you upgrade your Oracle Identity Manager environment to 11*g* Release 2 (11.1.2.3.0), the custom attributes for entities (such as users, roles, organizations, and application instances) exist in the back-end. However, if you want to display these attributes as form fields in the Oracle Identity Manager user interface, then you must customize the associated pages on the interface to add the custom form fields. To do so, use the links in the Upgrade region of Identity System Administration.

The Upgrade region contains the following:

- Upgrade User Form

  Use this page to create and manage custom form fields for the user entity.

- Upgrade Role Form

  Use this page to create and manage custom form fields for the role entity.

- Upgrade Organization Form

  Use this page to create and manage custom form fields for the organization entity.

- Upgrade Application Instances

  Use this page to create and manage custom form fields for the application instance entity.

For detailed information about upgrading Oracle Identity Manager to 11*g* Release 2 (11.1.2.3.0), see *Upgrade Guide for Oracle Identity and Access Management*.

3.2.2.6 Workflows

The Workflows region consists of the Approval Workflow Rules page. Use this page to manage approval workflow rules that determines whether or not request approval is required for an operation and which workflow is invoked for a specific operation.

See "Managing Workflows" on page 4-1 for more information.

Part II

Policy Administration

This part describes policy administration in Oracle Identity Manager, such as managing approval workflows and access policies.

It contains the following chapters:

4

Managing Workflows

Request generation and approval is governed by the following:

- Whether Oracle Identity Manager is running with or without workflows. By default, workflows are enabled in Oracle Identity Manager. For information about running Oracle Identity Manager without workflows, see "Running Oracle Identity Manager Without Workflows" on page 4-31.
- Approval workflow rules defined for the supported operations.

> **Note:** Approval policies have been deprecated in favour of workflow policies. Request generation and approval is governed by workflow policies, as described in this document.
>
> However, if you have upgraded Oracle Identity Manager from an earlier release, then approval policies continue to work as described in the following URL:
>
> `https://docs.oracle.com/cd/E40329_01/admin.1112/e27149/appr_policies.htm#OMADM2264`

This chapter describes request generation and approval in the following sections:

- Understanding Workflow Rules
- Configuring Approval Workflow Rules
- Managing Request Approval in an Upgraded Deployment of Oracle Identity Manager
- Moving Workflow Policies From Test to Production
- Running Oracle Identity Manager Without Workflows

4.1 Understanding Workflow Rules

Workflow rules determine the following:

- Whether or not approvals are required for an operation
- Which workflow must be invoked for a specific operation

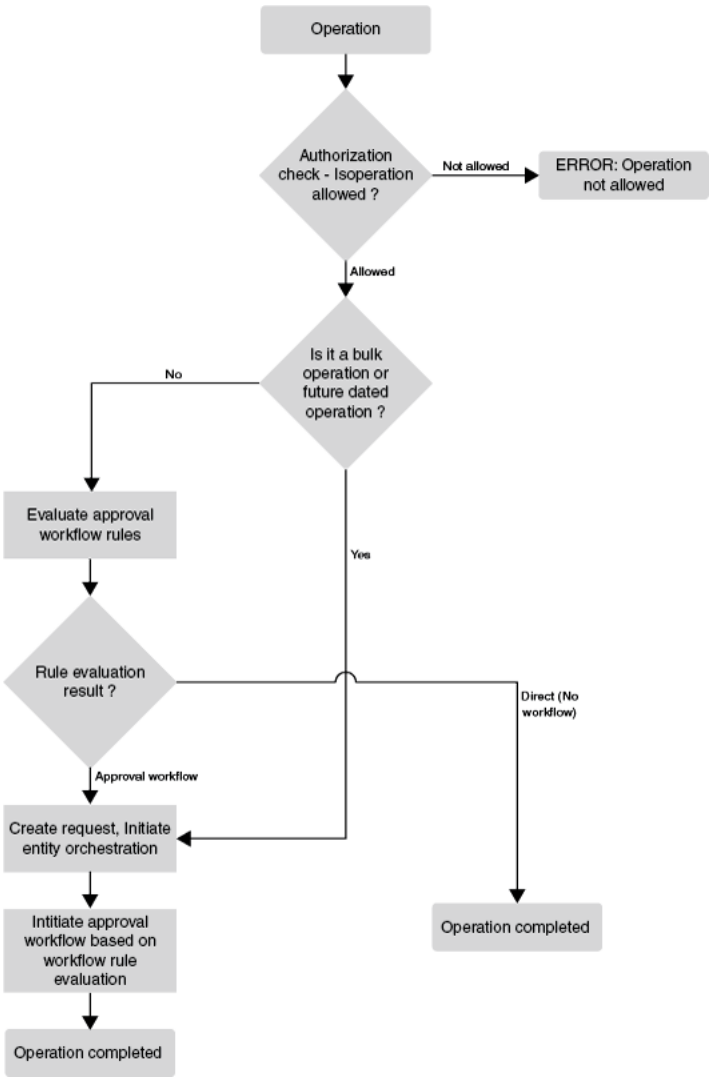
The following sections describe approval workflows:

- Request Process Flow
- Request Lifecycle

4.1.1 Request Process Flow

The process of request generation and approval, which is governed by approval workflow rules, is depicted in Figure 4–1.

Figure 4–1 Request Process Flow



The request process flow is as follows:

1. Authorization checks are performed based on the admin roles granted to the user. This check determines whether or not the user is allowed to perform the operation.
2. If the operation is not authorized, then an error is returned and the flow ends.
3. If the operation is allowed, then it is checked if the operation is a bulk operation or future-dated operation.
4. If it is a bulk or future-dated operation, then a request is created, approval is initiated based on workflow rule evaluation, and the operation is completed after approval.

5. If it is not a bulk or future-dated operation, then approval workflow rules are evaluated. The result of this evaluation determines whether request is created or not.
   - If the result is Direct, then no request is created, and the operation is performed directly.
   - If a SOA Workflow ID is returned as the result, then a request is created, and the workflow returned by the policy evaluation is invoked.

Bulk requests are processed in the following way:

- An allowed bulk operation always results in a request being created.
- Approval workflow rules configured for the bulk operation are evaluated.

  If rule evaluation results in a workflow ID, then bulk request is created and corresponding SOA workflow is initiated.

  If rule evaluation results in no workflow ID, then bulk request is created and it is auto-approved.
- After the bulk request is approved (auto-approved or SOA workflow approval), child requests are created.
- Child requests go through approval workflow rule evaluation (non-bulk), and are processed based on the outcome.

4.1.2 Request Lifecycle

Each request goes through a specific lifecycle after it is created in the system. The lifecycle transits the request through various stages. The stage a request is in determines what action the controller takes in that step, what operations are available on the request at that time, and what the possible stage transitions are.

Request lifecycle is described in the following sections:

- Request Stages
- Single Request Lifecycle
- Bulk Request Lifecycle

4.1.2.1 Request Stages

Table 4–1 describes how a request functions at various stages through its life cycle and how a request attains these stages.

Table 4–1 Request Stages

Request Stage	Description
Request Draft Created	After saving the request as a draft by clicking the **Save as Draft** button on the Cart Details page, the request moves to the Request Draft Created stage. A requester can save a request for modifying, submitting, or deleting it later. This is useful if the requester is awaiting additional information before submitting the request. The draft request cannot be withdrawn or closed. **Note:** The request data saved in draft mode does not include sensitive information such as passwords, even if they were entered before saving the request as draft.
Request Created	After successful submission of the request, the request moves to the Request Created stage.

Table 4–1 (Cont.) Request Stages

Request Stage	Description
Provide Information	This is a task assigned to requester (accessible from Inbox) for the entitlement request to search for and select the account for which the entitlement needs to be provisioned.
Request Awaiting Approval	After the request is created, the request moves to the `Request Awaiting Approval` stage automatically if there are approvals defined for this request. At this stage, the corresponding approvals are initiated through the request service. If a request is withdrawn or closed at this stage, then the request engine calls cancel workflow on each workflow instance. Notifications are sent to approvers about the withdrawn tasks. After the request successfully completes these stages, it will attain the Request Approved stage. If an SoD validation check is plugged-in after the request has been successfully created, the request is associated with the following stages. ■ SoD check not initiated A request attains this stage, if the SoD validation is not initiated for provisioning resource based request. The request engine moves the request to this stage after submission of request and before Obtaining Approval. ■ SoD check initiated A request attains this stage, if the SoD validation is initiated asynchronously for provisioning resource based request. The request engine moves the request to this stage after submission of request and before Obtaining Approval. ■ SoD check completed A request attains this stage, if the SoD validation is completed for provisioning resource based request. The request engine moves the request to this stage after submission of request and before Obtaining Approval. **Note:** These SoD request stages are possible if the request is any of the following request types: ■ Provision Application Instance ■ Modify Account ■ Provision Entitlement ■ Revoke Entitlement ■ Assign Roles ■ Remove from Roles
Request Approved	Only after a request is approved, it moves to the next stage and is updated with the current stage. The outcome is Approved, Rejected, or Pending.
Request Auto Approved	Only after a request is approved, it moves to the next stage and is updated with the current stage.
Request Rejected	Each time a workflow instance is updated, request service updates the request engine with the current stage of that instance. The outcome that the request engine expects from request service is Approved or Rejected. If any of the workflow instances that are instantiated are rejected, then request engine moves the request to Rejected stage. If any workflow instance is rejected, then the controller calls cancel on all the pending workflows and moves the request to Rejected stage.

Table 4–1 (Cont.) Request Stages

Request Stage	Description
Operation Initiated	After the request is approved, the request engine moves the request to the Operation Initiated stage and initiates the operation. The following request stages are associated with this stage: ■ Operation Completed After completing the actual requested operation, the request engine moves the request to the Operation Completed stage. This happens after Operation Initiated stage and is associated with Completed stage. ■ Post Operation Processing Initiated After the actual requested operation is completed, if there exists any additional operation that needs to be executed as post-processing, the request engine moves the request to the Post Operation Processing Initiated stage, before initiating those operations. This happens after Operation Completed stage. **Note:** In case of a bulk operation, child requests are created after request level approval, and the parent request moves to the "Request Awaiting Child Requests Completion" stage.
Request Failed	When the associated operations specified in the request fails to execute, the request cancels any pending operations and moves the request to the Request Failed stage. The following request stages are associated with this stage: ■ Request Failed When all associated operations specified in a request fail, the request is moved to the Request Failed stage. ■ Request Partially Failed When any associated operation specified in a request fails, the request is moved to the Request Partially Failed stage.
Request Withdrawn	A request can be withdrawn by the requester. At this stage, the request is associated to the Request Withdrawn stage, and the initiation of all approvals are canceled. **Note:** ■ A request can be withdrawn before Operation Initiated stage. After the request attains the Operation Initiated stage, the request cannot be withdrawn. ■ A request saved in draft mode cannot be withdrawn. ■ A request can always be withdrawn by a requester only, which is done by using Identity Self Service. ■ An administrator can close requests, which is similar to the withdraw function.
Request Closed	A request can be closed by the requester. At this stage, the request is associated to the Request Closed stage, and the initiation of all approvals are canceled. **Note:** ■ A request saved in draft mode cannot be closed. ■ An administrator can close requests, which is similar to the withdraw function.

Table 4–1 (Cont.) Request Stages

Request Stage	Description
Request Completed	After the execution of all operations specified in the request are completed, the request engine moves the request to the Request Completed stage. The following request stages are associated with this stage: ■ Request Completed with Errors A request attains this stage, when an actual requested operation executes fine, but fails to execute any of the post-processing operations. The Request Completed with Errors stage is associated with the Failed stage. ■ Request Completed A request attains this stage, when an actual requested operation executes fine without any errors. ■ Request Awaiting Completion When a request is scheduled to be executed on a future date, the request attains Request Awaiting Completion stage till the operation is completed on an effective date.

The successful attainment of a stage also results in the status of the request being updated to the corresponding status.

Operations can be executed manually or automatically by the system in response to an event. Examples of manual operations are:

- Save request as draft
- Edit/update draft request
- Submit request
- Close/cancel (withdraw) request
- Approve request when the service is notified that the approval workflow is successfully approved

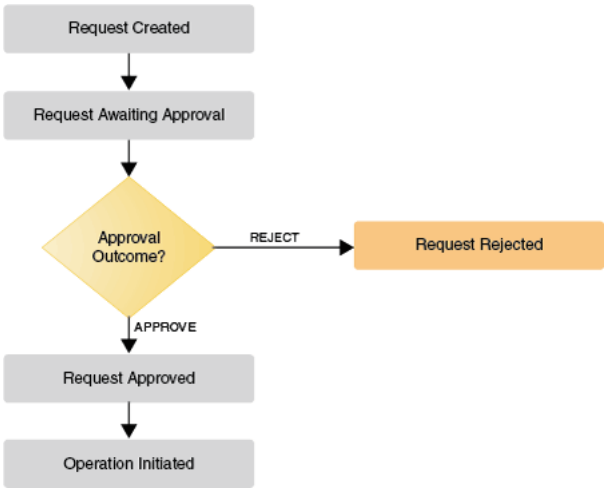
Examples of automatic operations are:

- Start approvals when the request is submitted
- Execute request when the request is approved and execution date is in the future or not specified

4.1.2.2 Single Request Lifecycle

Any single or non-bulk request goes through a single level of approval. When the approval workflow is invoked, the request moves to the `Request Awaiting Approval` stage, and it moves to the `Request Approved` stage after approval, as shown in Figure 4–2.

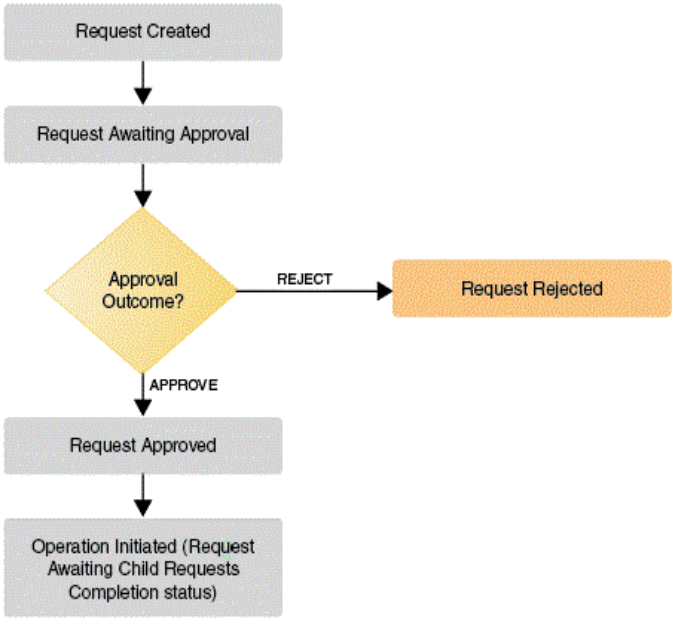
Figure 4–2 Single Request Lifecycle



4.1.2.3 Bulk Request Lifecycle

The lifecycle of a bulk or parent request is similar to a single or non-bulk request, until the bulk request goes to the `Request Approved` stage. After that, it is split into child requests, and then it moves to the `Request Awaiting Child Requests Completion` status of the `Operation initiated` stage, as shown in Figure 4–3.

Figure 4–3 Bulk Request Lifecycle



Each child request goes through the lifecycle described in "Single Request Lifecycle" on page 4-6. After the child requests are completed, the bulk or parent request moves to the `Request Completed` stage.

4.2 Configuring Approval Workflow Rules

Approval workflow rules can be configured to determine whether an operation requires approval or not. In addition, if approval is required, then the rule also indicates which SOA workflow is to be initiated.

A list of operations and corresponding workflow rules are predefined in Oracle Identity Manager. These system-defined rules determine whether or not approvals are required and the SOA workflow to be initiated. All the non-bulk operations have pre-defined approval workflow rules configured.

For a list of supported operations and corresponding rules, see "Understanding System-Defined Operations and Rules" on page 4-10.

Note: Oracle Identity Manager does not allow you to create new operations and corresponding rules. However, you can create and modify rules for the existing operations.

Approval workflow rules can be configured for all the supported operations, which are:

- Self-Register User
- Create User
- Modify User
- Disable User
- Enable User
- Delete User
- Create Role
- Modify Role
- Delete Role
- Assign Roles
- Remove from Roles
- Modify Role Grant
- Provision Application Instance
- Modify Account
- Disable Account
- Enable Account
- Revoke Account
- Provision Entitlement
- Modify Entitlement
- Revoke Entitlement
- Heterogeneous Request
- Bulk Modify User Profile
- Bulk Disable User

- Bulk Enable User
- Bulk Delete User
- Bulk Delete Role
- Bulk Assign Roles
- Bulk Remove from Roles
- Bulk Provision Application Instance
- Bulk Disable Account
- Bulk Enable Account
- Bulk Revoke Account
- Bulk Provision Entitlement
- Bulk Revoke Entitlement

Configuring approval workflow rules is described in the following sections:

- Understanding Rule Conditions
- Understanding System-Defined Operations and Rules
- Creating Approval Workflow Rules
- Configuring Custom Rule Conditions
- Modifying Approval Workflow Rules
- Deleting Approval Workflow Rules
- Understanding Approval Workflow Rule Evaluation

4.2.1 Understanding Rule Conditions

An approval workflow rule consists of:

- **Condition:** Rule condition based on the allowed inputs defined at the operation level
- **Outcome:** Workflow ID, which is the SOA workflow ID to be initiated for the operation

The following is an example of an approval workflow rule for the Modify User operation:

Rule condition:

```
requester.adminroles CONTAINS OrclOIMUserAdmin
```

Rule Outcome:

```
Direct
```

Here, the rule condition checks if the requester is a member of the User Administrator admin role in the beneficiary's organization. If the condition is satisfied, then operation is performed without initiating any approval workflow.

The rule conditions vary from operation to operation. For example, user data is required along with requester data for a Create User operation, and role information and user data is required along with requester date for an Assign Role operation. Requester data is required for all operations. Oracle Identity System Administration

enables you to enter the required role conditions based on the operation that you select.

Each approval workflow can have multiple rules associated with it, which must be defined in a certain order. For example, the Create User approval workflow can have rules, such as Create Contractor, Create Supplier, and Create Partner, defined in a sequence. The order in which the rules in an approval workflow are evaluated depends on the order or sequence in which the rules are defined in the policy.

See "Configuring Custom Rule Conditions" on page 4-16 for examples of rule conditions for each operation.

4.2.2 Understanding System-Defined Operations and Rules

Each operation/workflow policy has a default rule, whose outcome is DIRECT. This means that if the default rule condition evaluates to true for an operation, then it is a direct operation without approvals.

Table 4–2 lists the system-defined operations and corresponding workflow rules, for which the outcome is DIRECT.

Note: The rules in Table 4–2 are only for backward compatibility. You must remove these and create your own rules.

Table 4–2 Operations and Rules

Operation	Rule Name	Rule condition
Assign Roles	Assign Roles Default Rule	requester.adminroles CONTAINS OrclOIMRoleAuthorizer OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Create Role	Create Role Default Rule	requester.adminroles CONTAINS OrclOIMRoleAdministrator OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Create User	Create User Default Rule	requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Delete Role	Delete Role Default Rule	requester.adminroles CONTAINS OrclOIMRoleAdministrator OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Delete User	Delete User Default Rule	requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Disable Account	Disable Account Default Rule	requester.adminroles CONTAINS OrclOIMApplicationInstanceAuthorizerRole OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Disable User	Disable User Default Rule	requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Enable Account	Enable Account Default Rule	requester.adminroles CONTAINS OrclOIMApplicationInstanceAuthorizerRole OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator

Table 4–2 (Cont.) Operations and Rules

Operation	Rule Name	Rule condition
Enable User	Enable User Default Rule	requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Modify Account	Modify Account Default Rule	requester.adminroles CONTAINS OrclOIMApplicationInstanceAuthorizerRole OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Modify Entitlement	Modify Entitlement Default Rule	requester.adminroles CONTAINS OrclOIMEntitlementAuthorizer OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Modify Role	Modify Role Default Rule	requester.adminroles CONTAINS OrclOIMRoleAdministrator OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Modify User Profile	Modify User Profile Default Rule	requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Provision Application Instance	Provision ApplicationInstance Default Rule	requester.adminroles CONTAINS OrclOIMApplicationInstanceAuthorizerRole OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Provision Entitlement	Provision Entitlement Default Rule	requester.adminroles CONTAINS OrclOIMEntitlementAuthorizer OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Remove from Roles	Remove from Roles Default Rule	requester.adminroles CONTAINS OrclOIMRoleAuthorizer OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Revoke Account	Revoke Account Default Rule	requester.adminroles CONTAINS OrclOIMApplicationInstanceAuthorizerRole OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Revoke Entitlement	Revoke Entitlement Default Rule	requester.adminroles CONTAINS OrclOIMEntitlementAuthorizer OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator
Modify Role Grant	Modify Role Grant Default Rule	requester.adminroles CONTAINS OrclOIMRoleAuthorizer OR requester.adminroles CONTAINS OrclOIMUserAdmin OR requester.adminroles CONTAINS OrclOIMSystemAdministrator

In addition, there are a few other system-defined rules that support compliance use cases, which include role lifecycle, identity auditor, and certification, as listed in Table 4–3.

Table 4–3 Rules for Compliance Use Cases

Operation	Rule Name	Rule Condition	Rule Outcome
Assign Roles	Assign Roles IdentityAuditorEnabled Rule	identityAuditEnabled EQUAL TRUE	Workflow default/DefaultOperational Approval!5.0
Create Role	Create Role IdentityAuditorEnabled Rule	identityAuditEnabled Equal TRUE	Workflow default/RoleLCMApproval! 1.0
Delete Role	Delete Role IdentityAuditorEnabled Rule	identityAuditEnabled Equal TRUE	Workflow default/RoleLCMApproval! 1.0
Disable User	Disable User Certification Rule	request.isCertification Equal true	Workflow default/DefaultRequestAp proval!5.0
Modify Role	Modify Role IdentityAuditorEnabled Rule	identityAuditEnabled Equal TRUE	Workflow default/RoleLCMApproval! 1.0
Remove from Roles	Remove from Roles IdentityAuditorEnabled Rule	identityAuditEnabled Equal TRUE	Workflow default/DefaultOperational Approval!5.0
	Remove from Roles Certification Rule	request.isCertification Equal TRUE	Workflow default/DefaultRequestAp proval!5.0
Revoke Account	Revoke Account Certification Rule	request.isCertification Equal TRUE	Workflow default/DefaultRequestAp proval!5.0
Revoke Entitlement	Revoke Entitlement Certification Rule	request.isCertification Equal TRUE	Workflow default/DefaultRequestAp proval!5.0
Modify Role Grant	Modify Role Grant IdentityAuditorEnabled Rule	identityAuditEnabled EQUAL TRUE	Workflow default/DefaultOperational Approval!5.0

See Also: Table 4–4, " Approval Workflow Rule Syntax and Examples" for information about the `request.isCertification` and `identityAuditorEnabled` conditions.

Note: The workflow rules listed in Table 4–3 are configured ahead (in terms of order) of the default rules listed in Table 4–2. Therefore, these rules would be evaluated before the default rules. See "Understanding Approval Workflow Rule Evaluation" on page 4-24 for more information about workflow rule evaluation.

For example, the Assign Roles operation has two rules configured by default in the following order:

1. Assign Roles IdentityAuditorEnabled Rule
2. Assign Roles Default Rule

To determine the approval workflow to be initiated for an Assign Roles operation, the `Assign Roles IdentityAuditorEnabled Rule` rule is evaluated first. If the rule does not match (evaluates to true), then `Assign Roles Default Rule` is evaluated.

4.2.3 Creating Approval Workflow Rules

To create an approval workflow rule:

1. Login to Oracle Identity System Administration.
2. On the left navigation pane, under Workflows, click **Approval**. The Approval Workflow Configuration page is displayed.
3. In the Select an Operation to configure rules section, select an operation to configure rules for the operation. The rules associated with the operation are displayed in the Rules section at the bottom of the page. This section displays the rule name, rule description, and the workflow associated with the operation.

   In the Rules section, you can create a new rule, or select an existing rule and update or delete it.

   Note: When multiple rule conditions are specified in an approval workflow policy, the order in which rules are evaluated is based on the order in which they are configured in the policy. The order cannot be changed after the rules have been created. Therefore, the rules must be created in the order in which you want them to be evaluated.

4. In the Rules section, click **Create**. The Create Rule page is displayed.
5. In the Name box, enter a name for of the rule. This is a mandatory field.
6. In the Description box, enter a description for the rule.
7. In the Owner box, specify a owner of the rule. To do so, click the search icon adjacent to the Owner field, and then search and select a owner.
8. From the Status list, select a status for the rule. By default, the rule is in Enabled status.
9. In the Condition Builder section, specify the rule conditions in the IF and THEN clauses. To do so, perform the following steps to create a sample rule condition in which, if the user is a member of the Top organization, then the Create User operation is auto-approved:

a. Under the IF clause, in the first field of the empty row, to specify an object and attribute, click the search icon adjacent to the field. The Condition Builder dialog box is displayed.

Note: If you are aware of the exact object and attribute name, then you can enter the condition in the first field, for example `requester.Organization Name`, instead of clicking the search icon.

b. Click **requester** because you want to specify the condition based on requester data. A list of attributes is displayed that you can specify for the requester object. You can navigate through the attributes by clicking the page number icons and select it. Otherwise, enter the attribute name in the search field and click the search icon.

Note: From any screen of the Condition Builder dialog box, you can click **Start** to come back to the first screen in which you can start specifying a fresh condition by selecting the object.

c. Click the Organization Name attribute. The condition `requester.Organization Name` is displayed in the Condition Builder dialog box.

d. Click **OK**. The condition is added to the first field in the IF clause.

e. From the Operator list, select **Equal**.

The following operators are available for selection:

- EQUAL
- NOT_EQUAL
- CONTAINS
- DOES_NOT_CONTAIN
- BEGINS_WITH
- DOES_NOT_BEGIN_WITH
- ENDS_WITH
- DOES_NOT_END_WITH

f. To specify the value in the field on the right side, click the search icon. The Condition Builder dialog box is displayed.

Note: If you are aware of the exact value, then you can enter the value, for example `Top`, instead of clicking the search icon.

g. Select any one of the following options:

- **Value:** To specify the value of an attribute.
- **Expression:** To specify the condition based on an expression.

For the purpose of this example, select the **Value** option. The values for the Organization Name attribute are listed. This is because the object and attribute specified in the rule is `requester.Organization Name`.

h. Click **Top**. The Top organization is selected.

i. Click **OK**. The Top organization is populated in the value field.

j. To add another condition, click **Add Condition**. Another row is added under the IF clause. From the operator list on the right, you can select the **AND** or **OR** operator, and enter another rule condition as described in steps a through i.

   To remove a row, you can select the check box to the left of the row, and click **Remove**.

k. If you have added multiple rule conditions, then you can group the conditions together. To do so, select the check boxes to the left of the conditions, and click **Group**. Similarly, to remove the grouping of the conditions, select the check boxes to the left of the conditions, and click **Ungroup**.

Note:

- You can group only two conditions at a time. If you select more than two conditions, then the **Group** button is disabled. Alternatively, the **Ungroup** button is enabled only when you select one of the conditions that is grouped, but it is disabled when you select more than one group.
- A maximum of two conditions can be grouped together. Therefore, if you create a rule with four conditions that are grouped together with the AND operator, then the conditions are grouped into two sets. But if one of the conditions are grouped with the OR operator, then rule is updated correctly.

---

l. In the THEN clause, click the search icon adjacent to the first field to open the Condition Builder dialog box.

m. Select **workflow** and click **OK**.

n. In the value field for workflow, select **AutoApproval!1.0**. The request is auto-approved if you select this workflow.

o. Click **OK**. The workflow value is populated in the value field.

   Therefore, the rule condition you specified is the following:

```
IF
requester.Organization Name EQUALS Top

THEN
workflow default/AutoApproval!1.0
```

10. Click **Create** to create the rule condition. The rule condition is displayed in the table when you select the Operation for which it is created.

Note:

- When Risk attributes are used to define the conditions in a rule, for the rule to be evaluated correctly, the Risk Aggregation Job scheduled job must be run before the request is made.
- For application instances, there is no mechanism to filter out the attributes. All the attributes for application instances are displayed in the Condition Builder with which a rule can be written. For roles, select the role name to display the list of attributes for the role entities. You can select the asterisk (*) wildcard character to display the list of attributes.

See "Configuring Custom Rule Conditions" on page 4-16 for examples of rule conditions for each workflow operation.

4.2.4 Configuring Custom Rule Conditions

Table 4–4 describes how to specify custom rule conditions with examples.

Table 4–4 Approval Workflow Rule Syntax and Examples

Operation	Top-Level Attribute for Condition	Creating Workflow Rule Conditions (based on top-level attribute)
All operations including bulk	operation	This refers to the operation being performed currently, for example: `operation EQUALS Create User` **Note:** Such a condition can be used where the desired outcome is always `true`.

All operations including bulk and excluding self-register user	requester	This refers to the user profile attributes, roles, and admin role memberships of all the requesters. For example:
		`requester.Email CONTAINS @mydomain.com`
		`requester.adminRoles CONTAINS OrclOIMUserAdmin`
		
		This condition means if requester's email ID contains mydomain.com, and requester is a member of OrclOIMUserAdmin admin role.

Create User	user	This refers to all the user entity attributes, which can be specified by the requester while creating a user. For example: `user.Last Name EQUALS Doe`
Modify User	user	This refers to all the user entity attributes of the user being modified, which can be specified by the requester while modifying a user. For example: `user.Organization EQUALS Marketing`
Disable User, Enable User, Delete User	user	This refers to all the user attributes of the user being disabled, enabled, or deleted, which are currently set in the user's profile. For example: `existingUser.Organization EQUALS Marketing`

Table 4–4 (Cont.) Approval Workflow Rule Syntax and Examples

Operation	Top-Level Attribute for Condition	Creating Workflow Rule Conditions (based on top-level attribute)
Disable User, Enable User, Delete User	request	This refers to the request metadata, and the only allowed subattribute is `isCertification`. The only allowed values are true and false, which must be entered manually. This can be used to check if the current operation/request is a certification request. For example: `request.isCertification EQUAL true` **Note:** `request.isCertification` is used within default rules. It is not recommended to create custom conditions using `request.isCertification`.

Create Role	role	This refers to all the role entity attributes, which can be specified while creating a role. For example:
		`role.Name EQUALS ITAdmin`
		
		Because catalog metadata attributes can also be specified while creating the role, conditions are allowed based on catalog metadata attributes as well. For example:
		`role.catalog.Category EQUAL Role`

Create Role	identityAuditEnabled	This can be used to create condition based on the value of the `OIG.IsIdentityAuditorEnabled` system property. The only allowed values are TRUE and FALSE, which must be entered manually. For example: `identityAuditEnabled EQUALS TRUE` **Note:** `identityAuditEnabled` is used within default rules. It is not recommended to create custom conditions using `identityAuditEnabled`.

Modify Role	role	This refers to all the role entity attributes, which can be specified while modifying a role. For example:
		`role.Name EQUAL ITAdmin`
		
		Because catalog metadata attributes can also be specified while modifying the role, conditions are allowed based on catalog metadata attributes as well. For example:
		`role.catalog.Category EQUAL Role`

Modify Role	existingRole	This refers to all the role entity attributes, which are currently set for the role being modified. For example:
		`existingRole.DisplayName EQUALS IT Administrator`
		
		Because catalog metadata attributes can also be specified while modifying the role, conditions are allowed based on catalog metadata attributes as well. For example:
		`role.catalog.Category EQUAL Role`

Table 4–4 (Cont.) Approval Workflow Rule Syntax and Examples

Operation	Top-Level Attribute for Condition	Creating Workflow Rule Conditions (based on top-level attribute)
Modify Role	identityAuditEnabled	This can be used to create condition based on the value of the `OIG.IsIdentityAuditorEnabled` system property. The only allowed values are TRUE and FALSE, which must be entered manually. For example: `identityAuditEnabled EQUALS TRUE` **Note:** `identityAuditEnabled` is used within default rules. It is not recommended to create custom conditions using `identityAuditEnabled`.
Delete Role	existingRole	This refers to all role entity attributes, which are currently set for the role being deleted. For example: `existingRole.DisplayName EQUALS IT Administrator`
Delete Role	identityAuditEnabled	This can be used to create condition based on the value of the `OIG.IsIdentityAuditorEnabled` system property. The only allowed values are TRUE and FALSE, which must be entered manually. For example: `identityAuditEnabled EQUALS TRUE` **Note:** `identityAuditEnabled` is used within default rules. It is not recommended to create custom conditions using `identityAuditEnabled`.
Assign Roles, Remove from Roles	user	This refers to all user attributes, which are currently set in the profile for the user/beneficiary who is being assigned a role or whose role membership is being revoked. For example: `user.Organization EQUALS Marketing`
Assign Roles, Remove from Roles	role	This refers to all attributes of the role, which is being assigned ro or revoked from a user. For example: `role.name EQUAL IT Administrator`
Assign Roles, Remove from Roles	catalogItem	This refers to catalog metadata attributes corresponding to the role for which the access request is being submitted. For example: `catalogItem.Category EQUAL Role`
Assign Roles, Remove from Roles	identityAuditEnabled	This can be used to create condition based on the value of the `OIG.IsIdentityAuditorEnabled` system property. The only allowed values are TRUE and FALSE, which must be entered manually. For example: identityAuditEnabled EQUALS TRUE **Note:** `identityAuditEnabled` is used within default rules. It is not recommended to create custom conditions using `identityAuditEnabled`.

Table 4–4 (Cont.) Approval Workflow Rule Syntax and Examples

Operation	Top-Level Attribute for Condition	Creating Workflow Rule Conditions (based on top-level attribute)
Assign Roles, Remove from Roles	request	This refers to the request metadata, and the only allowed sub-attribute is `isCertification`. The only allowed values are true and false, which must be entered manually. This can be used to check if the current operation/request is a certification request. For example: `request.isCertification EQUAL true` **Note:** `request.isCertification` is used within default rules. It is not recommended to create custom conditions using `request.isCertification`.
Modify Role Grant	user	This refers to user attributes, which are currently set in the profile for the user/beneficiary whose role membership is being modified. For example: `user.Organization EQUALS Marketing`
Modify Role Grant	role	This refers to attributes of the role whose membership is being modified. For example: `role.name EQUALS IT Administrator`
Modify Role Grant	catalogItem	This refers to catalog metadata attributes corresponding to the role for which the access request is being submitted. For example: `catalogItem.Category EQUAL Role`
Modify Role Grant	identityAuditEnabled	This can be used to create condition based on the value of the `OIG.IsIdentityAuditorEnabled` system property. The only allowed values are TRUE and FALSE, which must be entered manually. For example: `identityAuditEnabled EQUALS TRUE` **Note:** `identityAuditEnabled` is used within default rules. It is not recommended to create custom conditions using `identityAuditEnabled`.
Provision ApplicationInstance	user	This refers to the user profile attributes of the user/beneficiary to whom the account is being provisioned. For example: `user.Organization EQUALS Marketing`

Table 4–4 (Cont.) Approval Workflow Rule Syntax and Examples

Operation	Top-Level Attribute for Condition	Creating Workflow Rule Conditions (based on top-level attribute)

Provision ApplicationInstance	appType	This refers to the account which is being provisioned to a user/beneficiary.
		`appType` is a top-level attribute that can lead to a hierarchy of sub-attributes, such as `appInstance`, followed by `account`, and optionally followed by account-specific child tables or entitlements. Further, `account` can be followed by the parent form attributes, and child table or entitlement can be followed by their specific attributes.
		Example 1:
		`appType[AD User].appInstance[VisionEmployeesDomain].account[*].Organization Name EQUAL Marketing`
		This condition means that if an account is being created in the `VisionEmployeesDomain` appInstance within the `Marketing` organization.
		Note: Workflow rule evaluation only considers the account that is being requested and does not consider any of the existing accounts that the user/beneficiary might have.
		Example 2:
		`appType[AD User].appInstance[*].account[*].Organization Name EQUAL Marketing`
		This condition means that if an account is being created in any appInstance pertaining to the AD User target within the `Marketing` organization.

Provision ApplicationInstance	catalogItem	This refers to catalog metadata attributes set in the catalog item for which the access request is being submitted, for example `catalogItem`.
Modify Account	user	This refers to user profile attributes of the user/beneficiary whose account is being modified. For example: `user.Organization EQUALS Marketing`
Modify Account	appType	This refers to the account information that is being modified as part of this operation. For example: `existingAppType[AD User].appInstance[*].account[*].Organization Name EQUAL Manufacturing` This condition means that if the user account on any of the AD User targets is being transferred to the Manufacturing organization.

Table 4–4 (Cont.) Approval Workflow Rule Syntax and Examples

Operation	Top-Level Attribute for Condition	Creating Workflow Rule Conditions (based on top-level attribute)
Modify Account	existingAppType	This refers to current or existing user account information, which is being modified as part of this operation. For example: `appType[AD User].appInstance[*].account[*].Organization Name EQUAL Marketing` This condition means that if the user account on any of the AD User targets is being transferred from the Marketing organization to some other organization. **Note:** Workflow rule evaluation only considers the account that is being modified and does not consider any of the existing accounts that the user/beneficiary might have.
Modify Account	catalogItem	This refers to the catalog metadata attributes set in the catalog item for which the access request is being submitted. For example: `catalogItem.Category EQUALS Role`
Enable Account, Disable Account, Revoke Account	user	This refers to the user profile attributes of the user/beneficiary whose account is being enabled/disabled/revoked. For example: `user.Organization EQUALS Marketing`
Enable Account, Disable Account, Revoke Account	existingAppType	This refers to current or existing user account information, which is being disabled/enabled/revoked as part of this operation. For example: `existingAppType[AD User].appInstance[*].account[*].Organization Name EQUAL Marketing` This condition means that if the user account being disabled/enabled/revoked belongs to the Marketing Organization of any of the AD User targets. **Note:** Workflow rule evaluation only considers the account which is being disabled/enabled/revoked and does not consider any of the existing accounts that the user/beneficiary might have.
Enable Account, Disable Account, Revoke Account	catalogItem	This refers to catalog metadata attributes set in the catalog item for which the access request is being submitted, for example `catalogItem`.
Enable Account, Disable Account, Revoke Account	request	This refers to the request metadata, and the only allowed sub-attribute is `isCertification`. The only allowed values are true and false, which must be entered manually. This can be used to check if the current operation/request is a certification request. For example: `request.isCertification EQUAL true` **Note:** `request.isCertification` is used within default rules. It is not recommended to create custom conditions using `request.isCertification`.

Table 4–4 (Cont.) Approval Workflow Rule Syntax and Examples

Operation	Top-Level Attribute for Condition	Creating Workflow Rule Conditions (based on top-level attribute)
Provision Entitlement	user	This refers to user profile attributes of the user/beneficiary to whom the entitlement is being provisioned. For example: `user.Organization EQUALS Marketing`
Provision Entitlement	appType	This refers to the entitlement information or the data that is being specified while performing the operation. For example: `appType[AD User].appInstance[VisionEmployeesDomain].account [*].UD_ADUSRC[*].Group Name EQUAL PasswordPolicyAdminGrp` This condition means that if the `PasswordPolicyAdminGrp` entitlement is being granted to the user/beneficiary on `VisionEmployeesDomain` application instance.
Provision Entitlement	catalogItem	This refers to the catalog metadata attributes set in the catalog item for which the access request is being submitted, for example, `catalogItem`.
Modify Entitlement	user	This refers to the user profile attributes of the user/beneficiary whose entitlement grant is being modified. For example: `user.Organization EQUALS Marketing`
Modify Entitlement	appType	This refers to the entitlement information or the data that is being specified while performing the operation. For example: `appType[AD User].appInstance[VisionEmployeesDomain].account [*].UD_ADUSRC[*].Group Name EQUAL PasswordPolicyAdminGrp` This condition means that if `PasswordPolicyAdminGrp` entitlement grant is being modified for the user/beneficiary on `VisionEmployeesDomain` application instance.
Modify Entitlement	existingAppType	This refers to the entitlement information or the existing entitlement form data, such as start date and end date. For example: `existingAppType[AD User].appInstance[VisionEmployeesDomain].account [*].UD_ADUSRC[*].Group Name EQUAL PasswordPolicyAdminGrp` This condition means that if `PasswordPolicyAdminGrp` entitlement grant is being modified for the user/beneficiary on VisionEmployeesDomain application instance.
Modify Entitlement	catalogItem	This refers to the catalog metadata attributes set in the catalog item for which the access request is being submitted, for example `catalogItem`.
Revoke Entitlement	user	User profile attributes of the user/beneficiary whose entitlement grant is being revoked. For example: `user.Organization EQUALS Marketing`

Table 4–4 (Cont.) Approval Workflow Rule Syntax and Examples

Operation	Top-Level Attribute for Condition	Creating Workflow Rule Conditions (based on top-level attribute)
Revoke Entitlement	existingAppType	This refers to the entitlement information or the existing entitlement form data, such as start date and end date. For example: `existingAppType[AD User].appInstance[VisionEmployeesDomain].account[*].UD_ADUSRC[*].Group Name EQUAL PasswordPolicyAdminGrp` This condition means that if `PasswordPolicyAdminGrp` entitlement grant is being modified for the user/beneficiary on `VisionEmployeesDomain` application instance.
Revoke Entitlement	catalogItem	This refers to the catalog metadata attributes set in the catalog item for which the access request is being submitted, for example `catalogItem`.
Revoke Entitlement	request	This refers to the request metadata, and the only allowed sub-attribute is `isCertification`. The only allowed values are true and false, which must be entered manually. This can be used to check if the current operation/request is a certification request. For example: `request.isCertification EQUAL true` **Note:** `request.isCertification` is used within default rules.It is not recommended to create custom conditions using `request.isCertification`.

4.2.5 Modifying Approval Workflow Rules

You can modify workflow rules to add, modify, or remove the rule conditions. To modify the workflow rule that you added in "Creating Approval Workflow Rules" on page 4-13:

1. On the left navigation pane of Identity System Administration, under Workflows, click **Approval**. The Approval Workflow Configuration page is displayed.
2. In the Operation section, select an operation to configure rules for the operation.
3. In the Rules section, click **Edit**. The Edit Rule page is displayed.
4. In the Operation section, search and select the operation for which you want to modify the workflow rule. For the purpose of this example, select **Create User**. The rules for the Create User operation is displayed in a table in the Rules section.
5. In the Rules section, select the rule that you want to edit, and click **Edit**. The Edit Rule page is displayed.
6. If you want to modify any rule attribute, then update the attribute values in the Edit Rule section.
7. In the Condition Builder section, you can modify an existing rule by modifying the object, attribute, or value fields. You can also add conditions to the existing ones, or remove rule conditions. Perform the following steps to add a rule condition that specifies that if the requester has the UserHelpDesk admin role, then the target user manager's approval is required for the Create User Operation:
   a. In the Condition Builder section, click **Add Condition**. A new row is added.
   b. From the operators list on the right, select **OR**.

c. In the first field of the row, specify `requester.adminRole` by using the Condition Builder dialog box. See step 10 of "Creating Approval Workflow Rules" on page 4-13 for information about selecting values in the Condition Builder dialog box.

d. From the operators list, select **CONTAINS**.

e. In the value field, select `OrclOIMUserHelpDesk` by using the Condition Builder dialog box.

f. In the THEN section, specify workflow and default/BeneficiaryManagerApproval!3.0 in the two fields respectively. Therefore, the complete rule condition is:

```
IF
requester.adminRoles CONTAINS OrclOIMUserHelpDesk

THEN
workflow default/BeneficiaryManagerApproval!3.0
```

This rule condition will ensure that if the requester has the UserHelpDesk admin role, then the target user manager's approval is required for creating the user.

8. Click **Update**. The workflow rule is updated with the new rule condition.
9. To remove a rule condition, select the check box to the left of the rule condition row, and click **Remove**. Then, click **Update**.

4.2.6 Deleting Approval Workflow Rules

You can delete the workflow rules that you define for all the operations. However, it is recommended that the default rules are not deleted.

To delete a workflow rule:

1. On the left navigation pane of Identity System Administration, under Workflows, click **Approval**. The Approval Workflow Configuration page is displayed.
2. In the Operation section, select an operation whose workflow rule you want to delete.
3. In the Rules section, click **Delete**. A message box is displayed asking for confirmation.
4. Click **Yes**.

4.2.7 Understanding Approval Workflow Rule Evaluation

When an operation (bulk or non-bulk) is being performed, approval workflow rule evaluation takes place in the following way:

1. The approval workflow rules associated with the operation being performed are evaluated one by one, in the order in which they are configured.
2. Rule evaluation stops, and the outcome, which is workflowID or Direct, of the matched rule is returned.

   Approval workflow rule evaluation stops at the first matching rule, which is the rule that evaluates to true, and that rule's outcome is returned as the result.

3. For a Bulk operation, if none of the rules match, then the SOA composite configured in `defaultRequestApprovalComposite` of SOAConfig is returned implicitly.
4. For a non-bulk operation, if none of the rules match, then the SOA composite configured in `defaultOperationApprovalComposite` of SOAConfig is returned implicitly.

If the approval workflow rule evaluation returns a WorkflowID, for example `UserManagerApproval`, then a request is created and the corresponding ASYNC orchestration is initiated. As part of the orchestration, there is a possibility that some of the data submitted by the user is modified or added. As a result, a different workflow ID than `UserManagerApproval` might be applicable. To handle such scenarios, approval workflow rules are re-evaluated before the workflow is initiated. If the re-evaluation results in a different workflowID, for example `HRManagerApproval`, then `HRManagerApproval` is initiated.

4.3 Managing Request Approval in an Upgraded Deployment of Oracle Identity Manager

In an upgraded deployment of Oracle Identity Manager, the approval workflow rules feature is disabled by default. As a result, the following occurs when an operation is initiated:

- Authorization policies and admin role assignments determines whether or not an operation requires approval, as described in section "9.4 Request vs. Direct Operation" in the *Performing Self Service Tasks with Oracle Identity Manager*.
- Approval policies are functional and determines which SOA workflow is to be invoked if approval is required.
- There are two levels of approval, and the functionality is as described in the following URL:

  `https://docs.oracle.com/cd/E40329_01/user.1112/e27151/req_mangmnt_user.htm#OMUSG191`
- If you enable workflow policies, then request generation and approval takes place in the same manner as in a fresh deployment of Oracle Identity Manager. However, you must migrate approval policies to workflow policies, as described in "Migrating Approval Policies to Approval Workflow Rules" on page 4-27.

> **Note:** After enabling workflow policies, you must not disable it again. Toggling between enabling and disabling workflows is not supported.

Most of the approval policy features can be achieved by using approval workflows. Table 4–5 lists the approval policy features that can be achieved by using approval workflows.

Table 4–5 Approval Policies to Approval Workflows

Approval policies	Approval workflow rules
Approval policies for a request type	Approval workflow rule specific to an operation
Approval policy level, which consists of request level and operation level	Single level of approval

Table 4–5 (Cont.) Approval Policies to Approval Workflows

Approval policies	Approval workflow rules
Scope type, scope	Approval workflow rule condition
Approval process configuration: auto approval	Approval workflow rule configuration: Select Direct
Approval process configuration: approval process	Workflow rule configuration: Search and select workflow
Approval policy rule	Approval workflow rule condition
Heterogeneous request type	Heterogeneous request policy
Bulk request type	Bulk policy for operation, for example Create User Bulk
Request level approval policies	Approval workflow rules
Operation level approval policy	NA
Priority	Policy order, in which the policies are configured
Hierarchical organization scoping	Policy condition based on organization hierarchy, for example: `user.Organization="VisionMarketing" OR user.parentOrganization="Vision" OR ...`

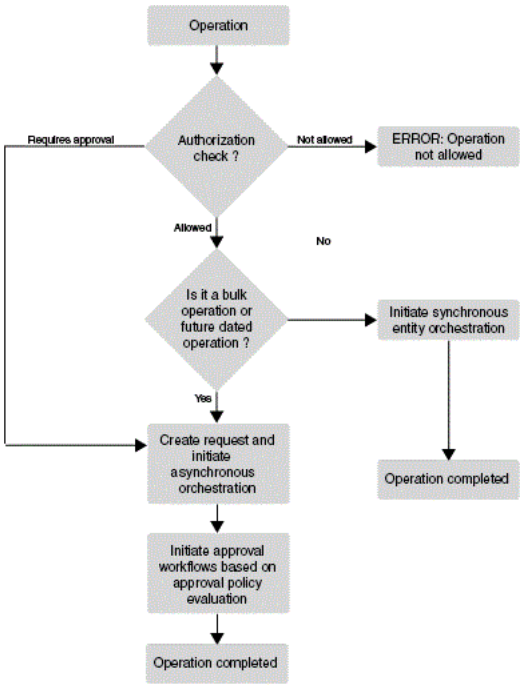
This section contains the following topics:

- Understanding Request Process Flow With Approval Workflow Rules Disabled
- Migrating Approval Policies to Approval Workflow Rules
- Enabling Approval Workflow Rules

4.3.1 Understanding Request Process Flow With Approval Workflow Rules Disabled

In an upgraded deployment of Oracle Identity Manager, approval workflows is disabled by default. Figure 4–4 shows the request process flow when approval workflows is disabled.

Figure 4–4 Request Process Flow with DIsabled Workflow



When approval workflow rules feature is disabled, the ApprovalRequired obligation(s) returned as a result of authorization policy evaluation determine whether the operation requires approval(s) or not.

If ApprovalRequired is false, then no request is created, and it is a direct operation.

If ApprovalRequired is true, then a request is created, approval policies are evaluated, and the SOA workflow returned by approval policy evaluation is initiated.

4.3.2 Migrating Approval Policies to Approval Workflow Rules

To migrate approval policies to approval workflow rules:

1. Identify all the approval policies that are applicable to a request type. Because there can be policies at request level and operation level, some manual analysis is required to identify their priority or order.

   An operation or request type can have multiple approval policies configured at request level and operation level. Whereas, by default, there is only a single approval workflow policy available for an operation. This default approval workflow policy cannot be deleted; new rules can be added to the same.

2. Pick the approval policy that comes first or next in the order of priority.
3. Open the default approval policy configuration specific to the request type. If there is a requirement to modify the current approval policy as a bulk workflow policy rule, then open the default bulk policy, for example Bulk Modify User.
4. Model the current approval policy as an approval workflow rule as follows:
   a. Create a new approval workflow rule with the same name as the approval policy name picked in step 2. Provide a description for the approval workflow rule.

See Also: "Creating Approval Workflow Rules" on page 4-13 and "Modifying Approval Workflow Rules" on page 4-23 for information about the user interface to work with approval workflow rules

b. In the Approval Workflow Configuration page of Oracle Identity System Administration, search and select the workflow that is configured in the approval policy as approval process.

c. Model the approval policy rule as approval workflow rule condition in the Approval Workflow Configuration page.

5. Repeat steps 2 through 4 for all the approval policies applicable to a request type.

6. Repeat steps 1 through 5 for all the request types.

4.3.3 Enabling Approval Workflow Rules

This section describes how to enable the approval workflow rules feature and the in-flight request lifecycle in an upgraded deployment of Oracle Identity Manager. It contains the following topics:

- Enabling the Approval Workflow Rules Feature
- Understanding In-Flight Request Lifecycle

4.3.3.1 Enabling the Approval Workflow Rules Feature

In an upgraded deployment of Oracle Identity Manager, the approval workflow rules feature is disabled by default. To enable the feature:

1. Ensure that SOA is enabled. To do so, verify that the value of the `Workflows Enabled` system property is `true`.

2. Ensure that migration of approval policies to approval workflows, as described in "Migrating Approval Policies to Approval Workflow Rules" on page 4-27, has been completed.

3. Set the value of the `Workflow Policies Enabled` system property to `true`.

4. Restart Oracle Identity Manager Managed Server.

4.3.3.2 Understanding In-Flight Request Lifecycle

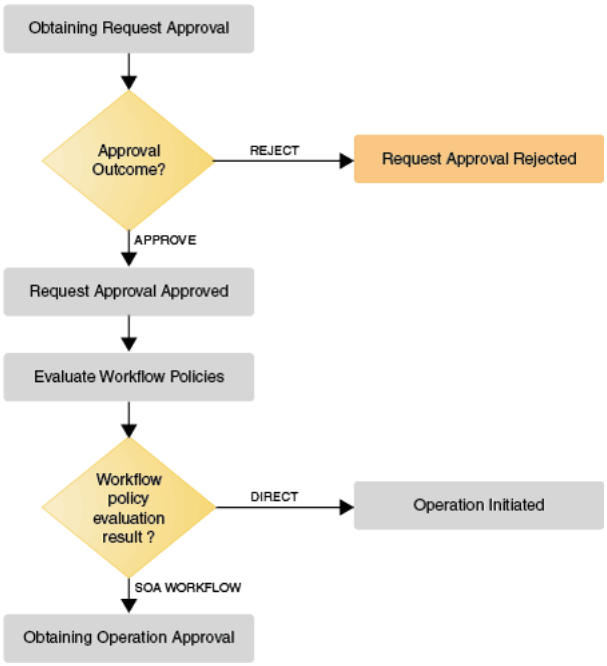
When you upgrade Oracle Identity Manager to 11*g* Release 2 (11.1.2.3.0), there can be some in-flight requests that must be processed after the upgrade. After the approval workflow policies feature is enabled, the life cycle of all the in-flight requests are the same as in 11g Release 2 (11.1.2.2.0), except for workflow determination. SOA workflow to be initiated is determined based on the workflow policies and not approval policies. In-flight request go through the existing request stages, which are Obtaining Request Approval, Obtaining Operation Approval, Request Approval Approved, Operation Approval Approved, Request Approval Rejected, and Operation Approval Rejected.

After enabling approval workflows, the in-flight requests are processed in the following manner:

For In-Flight Requests Awaiting Request Approval

Figure 4–5 shows the lifecycle of in-flight requests that are awaiting request approval.

Figure 4–5 In-Flight Requests Awaiting Request Approval



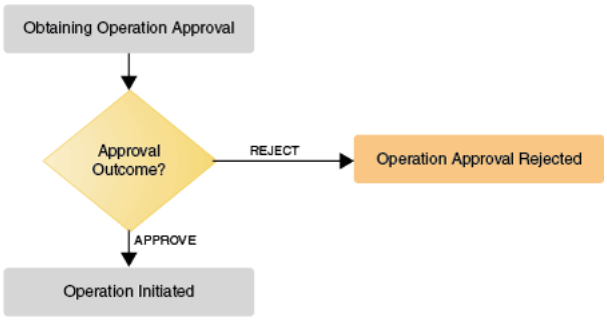
If the request is approved, then approval workflow rule is evaluated to determine the SOA workflow to be initiated at operation level. The request moves to the `Obtaining Operation Approval` stage.

If the request is rejected, then the request moves to `Request Approval Rejected` stage.

For In-Flight Requests Awaiting Operation Approval

Figure 4–6 shows the lifecycle of in-flight requests that are awaiting operation approval.

Figure 4–6 In-Flight Requests Awaiting Operation Approval



If the request is approved, then the operation is initiated.

If the request is rejected, then the request moves to the `Operation Approval Rejected` stage.

4.4 Moving Workflow Policies From Test to Production

Workflow rules can be exported from the source/test environment and imported to the target/production environment by using the Deployment Manager. Migration of workflow rules and rule to operation/policy relationships come under the category of Policy in the Deployment Manager Wizard. See "Migrating Incrementally Using the Deployment Manager" on page 21-1 for information about the Deployment Manager.

As workflow rules are associated with a specific operation, you must select the operation first, and then select the rules that you want to export.

While exporting/importing workflow rules, the workflow rule configuration in the source environment overrides the workflow rule configuration in the target environment. As a result, when workflow rules for an operation are imported, all rules configured for that operation are deleted, and the exported rules are associated with that operation in the target environment. In addition, the order of the rules in the source environment are carried over to the target environment.

For example, consider that the Create User operation has rules Rule1 and Rule2 configured in the target environment. But the Create User operation on the source environment has rules Rule1 and Rule3, and both are exported. When these rules are imported to the target environment, Rule1 and Rule2 are deleted, and Rule1 and Rule3 are associated with the Create User operation.

Therefore, it is recommended to maintain the source/test environment as the source of truth for workflow rule configuration.

To export/import the workflow rules by using the Deployment Manager:

1. Login to Oracle Identity System Administration of the source/test environment as the system administrator.
2. On the left pane, under System Configuration, click **Export**. The Deployment Manager opens and the Search Objects page of the Export Wizard is displayed.
3. From the drop down, select **Policy**, and then search for the operations and policies that you want to export.
4. In the search results, select the operation for which you wan to export the workflow rules, and then click **Select Children**.
5. In the Select Children page, select the workflow rules for the selected operations that you want to export.
6. Continue with the remaining steps of the wizard and complete the export.
7. Login to Oracle Identity System Administration of the target/production environment as the system administrator.
8. On the left pane, under System Configuration, click **Import**.
9. Select the file that contains the exported workflow rules (from step 6), and complete the import process.

Note: A workflow rule condition can refer to entities, such as role or application instance. Such dependent entities cannot be migrated as part of workflow rule migration. You must manually configure or migrate such dependent entities in the target/production environment. Otherwise, rule evaluation result might be unpredictable.

4.5 Running Oracle Identity Manager Without Workflows

Oracle Identity Manager is dependent on SOA server, which is installed and enabled by default. However, you can manually disable workflows by disabling SOA as a post install configuration step. This chapter describes the procedure to disable SOA and the functional impact of doing so in the following sections:

- Disabling SOA Server
- Understanding the Impact of Disabling Workflows

4.5.1 Disabling SOA Server

To disable SOA Server:

1. Shutdown the SOA Managed Server.
2. Set the value of the `Workflows Enabled` system property to `false`. See Table 20–1, " Default System Properties in Oracle Identity Manager" for information about this system property.
3. Restart Oracle Identity Manager Managed Server.

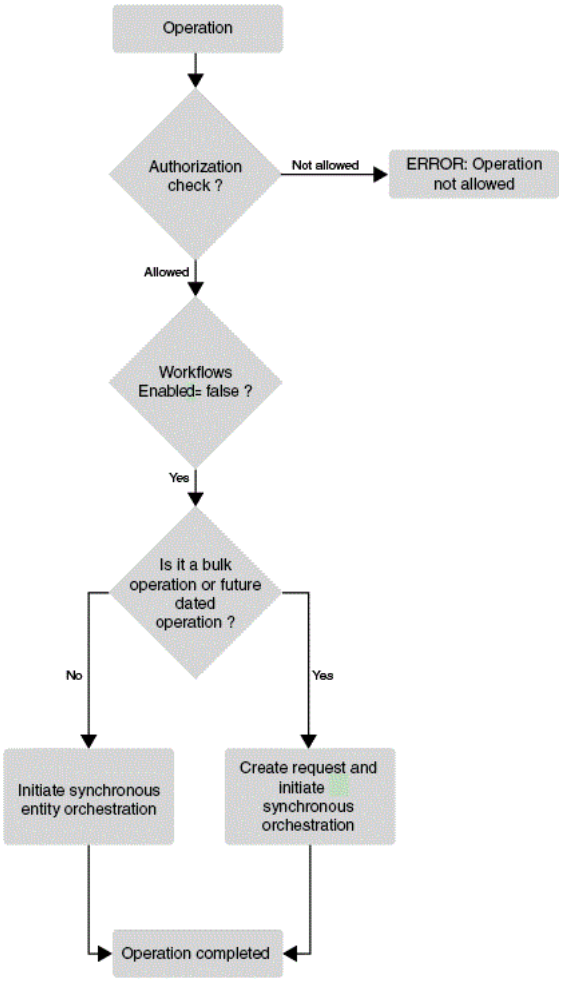
SOA Server can be re-enabled by setting the value of the `Workflows Enabled` system property to `true`.

Note: Oracle recommends that you do not enable SOA again after disabling it. Toggling between enabling and disabling workflows is not supported.

4.5.2 Understanding the Impact of Disabling Workflows

The primary functional impact of disabling workflow is that all operations are auto-approved, which means that the operations are completed without any approvals, as shown in Figure 4–7. Because no approval workflows are initiated, neither approval policies nor approval workflow rules are evaluated.

Figure 4–7 Disabled Workflows



In addition, Table 4–6 lists the features that are not available when workflow is disabled.

Table 4–6 Unavailable Features When Workflow is Disabled

Feature	Details
Request approvals	■ All the operations performed are direct operations and a request is not created, with the following exceptions: – Request is always created for bulk operations. Child requests are created immediately, which are auto-approved. – Request is always created for all operations with future effective date. All the newly created requests are auto-approved without involving any human approval. For example, all add access, self profile modification, and account/entitlement modification requests, are direct operations. ■ Approval policies or workflow rules are not evaluated. See "Understanding Rule Conditions" on page 4-9 for information about workflow rules.

Table 4–6 (Cont.) Unavailable Features When Workflow is Disabled

Feature	Details
Provisioning operations	■ **Disconnected application instance:** Manual fulfillment tasks for disconnected application instances do not work when workflow is turned off. Provisioning operations for disconnected application instances will fail. ■ **Account-entitlement dependency:** Entitlement request with one beneficiary when workflow is turned off works in the following way: – **Selected user has one account:** The account is preselected, and there is no impact. – **Selected user has multiple accounts:** It is mandatory to select an account, and there is no impact. – **Selected user has no account:** Application instance automatically gets added to the cart and a bulk request is created. As SOA is turned off, bulk and child requests is auto-approved. – **Selected user has a pending account request:** Newly created entitlement request is set as dependent on the account request. If the account is pending for approval, then as SOA is turned off, the requests are not processed further. If account request is waiting for an effective date, then the entitlement request is processed after the account request is completed. Entitlement request with multiple beneficiaries when SOA is turned off results in bulk request. The bulk request is auto-approved and child requests are created. The following are some specific use cases: – **User has one account:** entitlement request: This is auto-approved and completed. – **User has multiple accounts:** Corresponding entitlement child requests will fail. – **User has no account:** Corresponding entitlement child requests will fail. – **User has pending account request:** Entitlement child request is set as dependent on the account request. If the account is pending for approval, then as SOA is turned off, the requests are not processed further. If account request is waiting for an effective date, then entitlement request is processed after the account request is completed. The following account -entitlement use cases that rely on SOA composites will fail: – A multiple beneficiary request for entitlement where one or more beneficiaries have no account, and there is no in-flight account request (for that beneficiary and application instance combination). – A multiple beneficiary request for entitlement where one or more beneficiaries have multiple accounts, and account is not identified.

Table 4–6 (Cont.) Unavailable Features When Workflow is Disabled

Feature	Details
Identity Auditor features	When workflow is turned off, Certification and Identity Auditor features do not work, and the UI links related to Certification and Audit Compliance are not displayed in both Identity Self service and Identity System Administration. This is true in the following circumstances: ■ When the value of the `Display Certification or Attestation` system property is set to `Certification` or `Both`. ■ When the value of the `Identity Auditor Feature Set Availability` system property is set to `TRUE`. When workflow is turned off, any certification-related scheduled job will not be run, and appropriate messages is logged.
UMS notification	UMS notification provider works depending on whether workflows are enabled or not. UMS provider does not send notifications when SOA is not available. If UMS provider is kept enabled when SOA is not available, then UMS does not attempt to send notifications and logs an error message. An alternate provider, for example, the EmailServiceProvider must be configured and enabled to continue sending notifications.
Web services connector	Web services connector does not work when SOA is disabled.
SIL-based SoD	SIL-based SoD check do not happen when workflows are disabled even when the operation results in a request. However, SIL-based SoD Checks continue to work at the provisioning level when SOA is unavailable.
User management	■ **User self-registration:** User self-registration continue to work when SOA is turned off. Organization is calculated through the home organization determination policy, and the self registration request is auto approved. ■ **Proxy user management:** The proxy feature is disabled when SOA is turned off. The panel for managing proxies in the Identity Self Service is not displayed, and all the APIs around proxy throws an exception with the following message: `Proxy functionality is only supported when SOA and workflows are enabled.`
User interface	The following features are disabled (and not displayed) in the default user interface when the `Workflows Enabled` system property is set to false: ■ **In Oracle Identity System Administration:** The Approval Policies link ■ **In Oracle Identity Self Service:** The Certifications, My open tasks, and Pending Approvals links/icons in the Self Service Home page, and the Approvals tab in the Request Details/Summary pages

5

Managing Access Policies

Access policies are rules that are assigned to roles and dictate which target systems can be provisioned or deprovisioned to users to whom these roles are assigned. Essentially, it is an automated way of provisioning of target systems to users. This is explained with the help of the following example:

A user belongs to multiple roles created in Oracle Identity Manager. Suppose a role Vision North has a membership rule assigned to it. Membership rules can be designed based on the organization that the user belongs to, such as `Organization Name = "Vision North America"`. Roles can have access policies assigned to them. An access policy states which resource would be provisioned and/or denied to a role when the access policy is applicable. Therefore, when a user is created in the Vision North America organization, it satisfies a membership rule and grants the Vision North role to the user. This in turn triggers the access policy assigned to the role and then provisions or denies the resources mentioned in the access policy.

In this release of Oracle Identity Manager, you can search and then assign access policies to roles from the role wizard of Identity Self Service. You cannot assign roles to access policies in the access policy configuration section of the Identity System Administration.

This chapter describes how to create and use access policies for users and resources in Oracle Identity Manager. It contains the following sections:

5.1 Terminologies Used in Access Policies

The following terminologies are associated with access policies:

Access Policy Owner

In this release, access policy owner does not have any special privileges. As access policy configuration UI is available in the Identity System Administration, only system administrators can access this feature. Also there are no authorization checks added based on access policy owners in access policy management APIs.

Resource

A resource is a logical entity in Oracle Identity Manager that can be provisioned to a user or an organization in Oracle Identity Manager. For example, Microsoft Active Directory (AD), Microsoft Exchange, SAP, UNIX, and Database is modeled as a resource in Oracle Identity Manager.

Resources are templated definitions that are associated with one or more workflows called Provisioning Process in Oracle Identity Manager, which model the lifecycle management, such as how to provision, revoke, enable, and disable.

Resources also have entities called forms associated with them. Forms represent a collection of attributes associated with the resource. For instance, a form associated with AD server includes attributes such as SAM Account Name, Common Name, and User Principal Name. Forms also contain an attribute of type IT Resource (see "IT Resource Type" on page 5-2 for details).

Account

Accounts are actual instances of a resource that are created and provisioned to a user or organization in Oracle Identity Manager. For example, an e-mail account on an Exchange server is an account (instance) of resource type Exchange.

Accounts have specific values for the attributes of the associated form.

IT Resource Type

IT resource type is a logical entity in Oracle Identity Manager used to model a physical target and all its attributes including (but not limited to) the connectivity information and the credentials required to connect to the physical computer. For example, IT resource type AD server is used to model an actual AD server.

IT Resource Instance

These are actual instances of specific IT resource type that represent the actual physical target. They also have specific values for all the attributes of the physical target, such as IP address, port, user name, and password. Two physical AD servers in a deployment are represented by two instances of IT resource type AD Server.

Providing the IT resource instance in access policy defaults for parent data is mandatory.

Account Discriminator

Account discriminator is a collection of attributes on a form that uniquely identifies the logical entity on which accounts are created. This term is sometimes loosely referred to as a target. For instance, for an AD server, an account discriminator can be a combination of AD server (an attribute of type IT Resource) and Organization Name.

Typically, account discriminators are attributes of type IT Resource.

Attributes are marked as account discriminators by setting the Account Discriminator property of a Form field to True.

5.2 Features of Access Policies

This section describes the various features offered by the policy engine in the following sections:

- Direct Provisioning
- Revoking or Disabling the Policy

- Denying a Resource
- Evaluating Policies
- Evaluating Policies for Reconciled and Bulk Load-Created Accounts
- Access Policy Priority
- Access Policy Data
- Provisioning Multiple Instances of the Same Resource via Access Policy by Using Account Discriminator

5.2.1 Direct Provisioning

Whenever an access policy is applied, the resources are directly provisioned/denied without any request being generated.

5.2.2 Revoking or Disabling the Policy

Oracle Identity Manager access policies are not applied to subroles. Policies are only applied to direct-membership users (that is, users who are not in subroles) in the roles that are defined on the access policies. You must specify whether a resource in a policy must be revoked or disabled when the policy no longer applies. Based on your selection, the resources are automatically revoked from the users or disabled when the policy no longer applies to the users. Accounts and entitlements can either be revoked or disabled if policy no longer applies.

For each resource associated with an access policy, you must select any one of the following options:

- **Revoke if no longer applies:** Selecting this option revokes the account and the entitlements associated with the access policy when the access policy is no longer applicable.
- **Disable if no longer applies:** Selecting this option disables the account and the entitlements associated with the access policy when the access policy is no longer applicable.

When the **Revoke if no longer applies** option or the **Disable if no longer applies** option is selected, entitlements are always revoked with the policy no longer applies. If the **Disable if no longer applies** option is selected, then the entitlements associated with the resource are revoked when the policy no longer applies because the entitlements have been originally granted because of the role grant. The entitlements are added to the resource instance when the role is granted once again.

Note: During an upgrade to Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0), policies which had the Revoke if no longer applies option deselected is converted to Disable if no longer applies. Users associated with these policies will not be updated, but any future updates to the policy will result in the user being marked with a Disable if no longer applies flag.

If two policies have the same resource in the policy definition with one having the **Revoke if no longer applies** option selected and the other one with the **Disable if no longer applies** option, then the **Disable if no longer applies** option takes precedence over the **Revoke if no longer applies** option. In other words, resources are disabled (and not revoked) when both the policies no longer apply.

5.2.3 Denying a Resource

While creating an access policy, you can select resources to be denied along with resources to be provisioned for roles. If you first select a resource for provisioning and then select the same resource to be denied, then Oracle Identity Manager removes the resource from the list of resources to be provisioned. In other words, the resources to deny takes precedence.

Note: If a resource is denied by an access policy, then the resource is always denied, even if a different policy provisions it. Denying of resources is irrespective of access policy priority. Even if an access policy with lower priority denies a resource, it takes precedence over an access policy with higher priority.

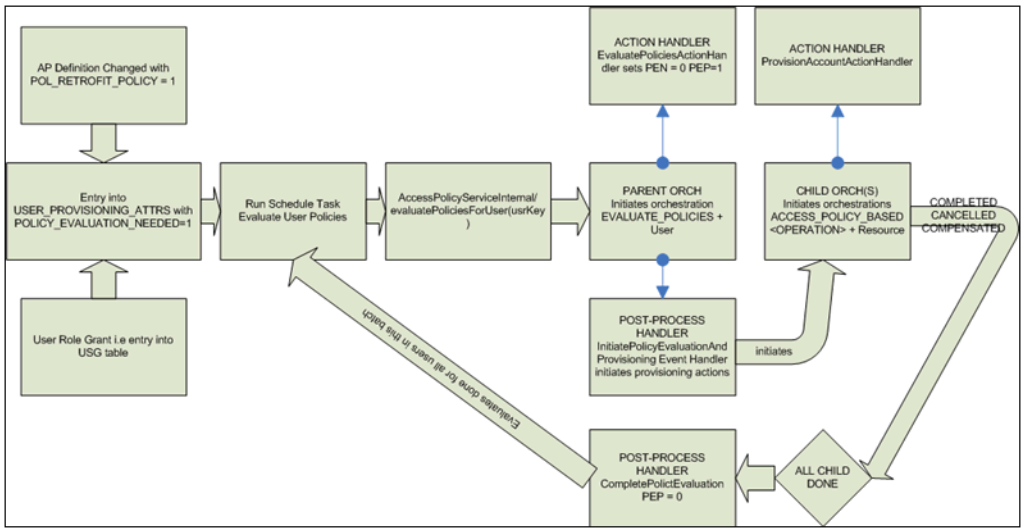
5.2.4 Evaluating Policies

Access policy evaluation works in the following manner:

1. After role grant to user or update of policy definition, the users for whom policy evaluation is needed are identified and the entry is updated or added in the USER_PROVISIONING_ATTRS table.
2. Flag stored in USER_PROVISIONING_ATTRS. POLICY_EVAL_NEEDED is used to determine users for whom policy evaluation is required.
3. The `Evaluate User Policies` scheduled job is run to invoke policy evaluation and further actions on accounts.By default, this task is enabled and scheduled to run every 10 minutes. See "Predefined Scheduled Tasks" on page 18-6 for more information about the Evaluate User Policies scheduled task.
4. The scheduled task picks distinct user records (each record in this table is for distinct user) available in USER_PROVISIONING_ATTRS in the default batch size of 500 using 20 threads, which is 25 records per thread in the ascending order of update date. A JMS message is submitted for each user for access policy evaluation.

Access policy evaluation is shown in Figure 5–1.

Figure 5–1 Access Policy Evaluation



When the policy is created with `Retrofit = true` and the policy definition is changed by one of the following manner:

- When a user is made a part of a role or removed from a role. The policy for the user is evaluated as part of the add or remove operation.
- If the retrofit flag is set for the policy. The evaluations can happen in the following scenarios:
  - Policy definition is updated. Policies are evaluated for all applicable users.
  - A role is added or removed from the policy definition. Policies are evaluated only for roles that is added or removed.

    Note: In 11*g* Release 2 (11.1.2.3.0), after the role is applicable to the user, you must run the Evaluate User Policies scheduled job to make access policy applicable.

  - A resource is added, removed, or the **Revoke If No Longer Applies** or **Disable If No Longer Applies** options are changed for the resource.

    When you change the **Revoke if no longer applies** and **Disable if no longer applies** options from one to the other, the existing resource instances are not re-evaluated immediately. This policy change takes effect only the next time the policy is evaluated by the Evaluate User Policies scheduled task.
  - When policy data is updated or deleted. This includes both parent and child form data. Policies are evaluated for all applicable users.

When roles are assigned to or removed from a policy, then users are immediately marked for policy evaluation irrespective of the setting for `Retrofit` flag. The `Retrofit` flag can be used only to determine if users need to be marked for policy evaluation for access policy definition changes, which involves:

- Priority of the policy changes
- Resource associated to policy changes
- The the **Revoke If No Longer Applies** or **Disable If No Longer Applies** options change
- Parent or child data changes. In this scenario, if `Retrofit` is set to `false`, then users are not marked for policy evaluation although policy definition has changed.

5.2.5 Evaluating Policies for Reconciled and Bulk Load-Created Accounts

The access policy engine can link access policies to reconciled accounts and to accounts created by the Bulk Load Utility. The access policies are then evaluated using the `Evaluate User Policies` scheduled job. To enable this feature, ensure the following:

- Set the values of `XL.AllowAPHarvesting` and `XL.AllowAPBasedMultipleAccountProvisioning` system properties to `TRUE`.

  See "System Properties in Oracle Identity Manager" on page 20-1 for information about these system properties. See "Creating and Managing System Properties" on page 20-22 for information about setting the value of a system property.
- Set the retrofit flag to ON for the policy to be linked by selecting Retrofit Access Policy.

For information about the retrofit flag, see "Creating Access Policies" on page 5-8. For information about updating a policy definition, see "Managing Access Policies" on page 5-11.

- Populate the ITResource field in access policy defaults.

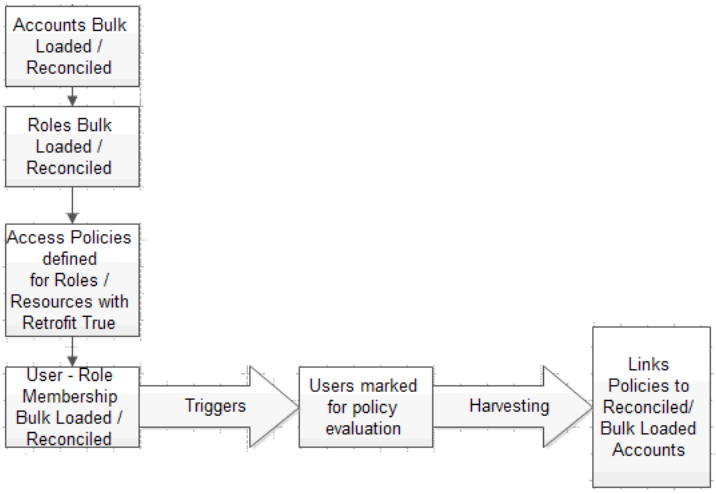
  If the ITResource field is not populated, then the provisioning engine cannot determine the application instance to be associated with the account and the account will not be visible in the UI.

- Designate a field on the process form as the discriminator field and set the value of the **Account Discriminator** property to `True`. Then, populate the access policy defaults for the account discriminator field.

  For information about setting the discriminator field, see "Enabling Multiple Account Provisioning" on page 5-12.

After enabling the feature to link access policies to reconciled accounts and to accounts created by the Bulk Load Utility, the access policy harvesting flow is depicted in Figure 5–2.

Figure 5–2 Access Policy Harvesting Flow



5.2.6 Access Policy Priority

Policy priority is a numeric field containing a number that is unique for each access policy you create. The lower the number, the higher is the priority of the access policy. For example, if you specify Priority =1, it means that the policy has the highest priority. When you define access policies through Oracle Identity System Administration, the value 1 is always added to the value of the current lowest priority and the resultant value is automatically populated in the Priority field. Changing this value to a different number might result in readjusting the priority of all the other access policies, thus ensuring that the priorities remain consistent. The following actions are associated with the priority number:

- If the priority number entered is less than 1, then Oracle Identity Manager will change the value to 1 (highest priority).
- If the priority number entered is greater than M, in which M is the current lowest priority, then Oracle Identity Manager will force to specify the value as less than or equal to M+1.

- Two access policies cannot have the same priority number. Therefore, assigning an already existing priority number to an access policy will lower the priority by 1 for all policies of lesser priority.

Conflicts can arise from multiple access policies being applied to the same user. Because a single instance of a resource is provisioned to the user through access policies, Oracle Identity Manager uses the highest priority policy data for a parent form. For child forms, Oracle Identity Manager uses cumulative records from all applicable policies.

If there is more than one access policy created for the same resource but granting different sets of entitlements and having different behavior when policy no longer applies, then the access policy with **Disable if no longer applies** (DLNA) option enabled has the highest priority irrespective of the access policy priorities. For example, if there is an access policy with **Revoke if no longer applies** (RLNA) option enabled and another policy with DLNA option enabled, then the policy with DLNA option enabled has higher precedence. Irrespective of the order in which the policies are applied, if only the policy with DNLA no longer applies but the other policy with RNLA still applies, then the resource would remain in the `Provisioned` status. Only after all access policies for that resource are no longer applicable, then the resource would move to the `Disabled` status.

If a policy with Disable if no longer applies is later converted to Revoke if no longer applies, then existing accounts that are associated with the policy are not updated to RLNA. The change is effective only for accounts to be created in future.

If a policy with RLNA is later converted to DLNA, then the accounts that are already revoked are not impacted. The change is effective only for accounts currently associated with this policy or accounts to be created in the future.

5.2.7 Access Policy Data

There are multiple ways in which process form data is supplied for resources during provisioning. The following is the order of preference built into Oracle Identity Manager:

1. Default values from the form definition using Oracle Identity Manager Design Console
2. Prepopulate adapters
3. Access policy data if resource is provisioned because of a policy
4. Data updated by Process Task

If a given option is available, then the rest of the options that are at a lower order of preference are overridden. For example, if Option 4 is available, then Options 3, 2, and 1 are ignored.

5.2.8 Provisioning Multiple Instances of the Same Resource via Access Policy by Using Account Discriminator

In earlier releases of Oracle Identity Manager, access policies can be used to manage only a single account for a resource object. In other words, if you already have resource provisioned to user (account has been created in the target system) and if another instance of the same resource is to be provisioned to the same user via access policy, then it is not possible in earlier releases of Oracle Identity Manager. To achieve the functionality of provisioning multiple instances of resource to a user, prior to access policy enhancement in Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0), you

must clone the connector that represents the target system in Oracle Identity Manager. Cloning of connector was error prone needed lot of effort for testing/maintenance of cloned resource. Access Policy enhancement done for provisioning of multiple instances of resource in Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) saves the time and effort on cloning connectors.

A target system, such as UNIX server, Active Directory (AD) server, database, SAP, or JD Edwards, is the external system to Oracle Identity Manager that must be provisioned to users in Oracle Identity Manager. The target system is represented by an entity called resource in Oracle Identity Manager. The server on which target system is installed is represented by IT resource in Oracle Identity Manager. And the login credentials provided to user accessing this target system is represented by an account in Oracle Identity Manager. A user can have multiple accounts on a single target system. For example, one account can be a service (administrator) account and another a regular account. Therefore, it is mandatory to have two accounts for a same user in a single target system. In addition, it is possible to have different instances of target system, such as multiple UNIX servers, database servers, and AD servers. As a result, it is required to create accounts on each instance of the target system for the same user. For implementation details, see "Creating Separate Accounts for the Same User and Same Resource on a Single Target System" on page 5-13.

In Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0), access policies can provision multiple accounts in the same target system as well as a single account in multiple instance of the same target system. While evaluating access policies and provisioning resources to user, Oracle Identity Manager checks if the resource has already been provisioned to the user or not. This is determined by checking the resource key (OBJ_KEY) of the resource provisioned to user. To have multiple instances to be provisioned through access policy, another criteria called *account discriminator* along with OBJ_KEY is required to distinguish the multiple instances of the same resource. Therefore, access policy checks the resource key as well as account discriminator to decide if the resource has been provisioned or not.

The account discriminator is a field on a process form (account data) that distinguishes two accounts of the same user, which can be present on the same target system or different target systems. For example:

- If user Jane.Doe is to be provisioned two accounts on two different UNIX servers, then IT resource can be used as account discriminator.
- If user John.Doe is to be provisioned two accounts on the same database instance, then distinct login IDs can be used as account discriminator.

> **See Also:** "Provisioning Multiple Instances of the Same Resource via Access Policy" on page 5-12 for the steps to provision data from multiple target systems

5.3 Creating Access Policies

You can define an access policy for provisioning resources to users who have roles defined in the policy by using the Access Policy Wizard.

Note:

- Identity Audit policies (SoD) are not evaluated during the creation of an access policy.
- The association of a role to an access policy is done as part of role management and not via the access policy UI.

To create an access policy:

1. Login to Oracle Identity System Administration.
2. Under Policies, click **Access Policies**. The Manage Access Policies window is displayed.
3. Click **Create Access Policy** to open the access policy creation page.
4. Enter information in the required fields indicated with an asterisk (*), such as access policy name and description.

   Note: The following special characters are not allowed in the access policy name:

   Semicolon (;)

   Hash (#)

   Percentage (%)

   Equal to (=)

   Bar (|)

   Plus (+)

   Comma (,)

   Forward slash (/)

   Back slash (\)

   Single quote (')

   Double quote (")

   Less than (<)

   Greater than (>)

5. From the Policy Owner list, select **USER** or **ROLE** as the policy owner type. Based on your selection, click the lookup icon and select a user or role as the policy owner.

   If you do no select a policy owner type, then the access policy is created with the default policy owner type as USER and policy owner as the logged-in user.

   If you select a policy owner type and do not select a policy owner value, which is USER or ROLE, and click **Continue**, then a validation error is displayed that does not allow you to proceed to the next screen.

6. Select **Retrofit Access Policy** to retrofit this access policy when it is created.

   If you select the **Retrofit Access Policy** option, then the access policy is applied to all the roles to which this access policy is associated. For information about assigning access policies to roles, see "The Access Policies Tab" in *Performing Self Service Tasks with Oracle Identity Manager*.

   If you do not select this option, then existing role memberships are not taken into consideration.

7. Click **Continue**.

   The Create Access Policy - Step 2: Select Resources (to provision) page is displayed.

8. Specify the resource to be provisioned for this access policy.

   Search for resources by using the filter search menu.

   - Select the name of the resource from the results table, and then click **Add**.
   - The names of the desired resources to provision appear in the Selected list. If you want to create an access policy that only denies resources, click **Continue** without selecting a resource.
   - To unassign the selected resources, select the resource in the Selected list and click **Remove**.

9. Click **Continue**.

   If there is a form associated with this resource, then the subsequent pages display the required fields. Enter the process details for each resource you selected. Note that the itres field is a mandatory field, and you must specify a value for the IT resource. Without IT resource being populated, the following error message is displayed:

   ```
   Either values for some required fields are missing or a value for a required
   field is invalid.
   ```

   If you do not want to edit the form fields, then click **Skip This Step**. The Create Access Policy - Step 2: Select Revoke or Disable Flag page is displayed.

   Note: Oracle recommends that you do not specify policy defaults for passwords and encrypted attributes.

10. For each resource listed in the page, select any one of the following options:

    - **Revoke if no longer applies:** Selecting this option revokes the account and the entitlements associated with the access policy when the access policy is no longer applicable.
    - **Disable if no longer applies:** Selecting this option disables the account and the entitlements associated with the access policy when the access policy is no longer applicable.

    See "Revoking or Disabling the Policy" on page 5-3 and "Evaluating Policies" on page 5-4 for more information about revoking or disabling accounts and entitlements when access policies no longer apply.

11. Click **Continue**.

    The Create Access Policy - Step 3: Selected Resources (to deny) page is displayed.

12. Use this page to select resources to be denied by this access policy.

    To select resources to be denied:

    a. Select the resources from the results table.

    b. Click **Add** to place the resource in the Selected list.

       You must select at least one resource to deny if you have not selected any resources to be provisioned. Selecting the same resources to be denied as to be provisioned will automatically unassign them from the resources to be provisioned selection.

       Similarly, in Step a, assigning the same resources to be provisioned as you have already selected to be denied will automatically remove them from the

resources to be denied selection. You can remove the resources that were selected to be denied. You do this by selecting those resources from the **Selected** list, and clicking **Remove**.

c. Click **Continue**.

The Create Access Policy - Step 5: Verify Access Policy Information page is displayed.

13. If you want to modify any of the selections you made in the preceding steps of this procedure, then click **Change** to go to the corresponding page of the wizard. After making the required modifications, click **Continue** to return to the Step 5: Verify Access Policy Information page.

Note: The Policy Owner Type and Policy Owner fields are displayed only if values for these fields have been specified, as described in step 5.

14. Click **Create Access Policy** to create the access policy.

Note: When you create an access policy on a resource having a process form with Password field, the password policy is not evaluated. For information about password policies, see .

5.4 Managing Access Policies

You can use Oracle Identity System Administration to modify information in existing access policies.

To manage access policies:

1. In the Identity System Administration, click **Access Policies** under Policies.

   The Manage Access Policies page is displayed.

   Use the menu in the search criteria field to select an access policy attribute. You can use the asterisk (*) wildcard character to search for all access policy instances that have any value for the attribute selected. Click **Search Access Policies**.

   The Manage Access Policies page is displayed with your search results.

2. To view the details of the Access Policy you want, click the access policy name.

   The access policy details page is displayed. To make modifications to this access policy, use the **Change** link at the end of each selection category. After you make the required modifications, click **Update Access Policy** to save the changes.

   The Policy Owner Type and Policy Owner fields are displayed only if values for these fields have been specified when creating the access policy. You can modify the values of both the Policy Owner Type and Policy Owner fields. If you change the value of the Policy Owner Type field from USER or ROLE to **Select Policy Owner Type**, then the existing policy owner type and policy owner is retained.

Note:

- You can change the **Revoke if no longer applies** and **Disable if no longer applies** options as a part of the access policy modification. See "Revoking or Disabling the Policy" on page 5-3 and "Evaluating Policies" on page 5-4 for information about the effects of changing these options in access policies.
- The IT Resources field cannot be edited from the access policy details page.

3. After you complete making the required modifications for the selection categories, click **Exit**.

See Also: "The Access Policies Tab" in Performing Self Service Tasks with Oracle Identity Manager for information about adding access policies to a role and removing access policies assigned to a role.

5.5 Provisioning Multiple Instances of the Same Resource via Access Policy

Provisioning multiple instances of the same resource via access policy by using account discriminator involves the following:

- Enabling Multiple Account Provisioning
- Creating Separate Accounts for the Same User and Same Resource on a Single Target System
- Provisioning Multiple Instances of a Resource to Multiple Target Systems
- Limitation of Provisioning Multiple Instances of a Resource via Access Policy

5.5.1 Enabling Multiple Account Provisioning

By default, Oracle Identity Manager does not support multiple account provisioning. To enable multiple account provisioning:

Set the value of the XL.AllowAPBasedMultipleAccountProvisioning system property to `TRUE`. See "System Properties in Oracle Identity Manager" on page 20-1 for information about this system property. See "Creating and Managing System Properties" on page 20-22 for information about setting the value of a system property.

When multiple account provisioning is enabled, you must define the appropriate account discriminator attributes. To do so:

1. Log in to the Design Console.
2. Update the process form as follows:
   a. Expand Development Tools, and then double-click **Form Designer**.
   b. Search and open the process form.
   c. On the Form Designer tab, click **Create New Version**.
   d. In the Create a New Version dialog box, enter a label in the Label field, and then click **Save**.
   e. From the Current Version list, select the version that you created.

f. On the Properties tab, select the field that you want to designate as the discriminator field, and then click **Add Property**.

g. In the Add Property dialog box, select **Account Discriminator** as the property name, enter `True` in the Property Value field, and then click **Save**.

h. Click **Make Version Active**, and then click **OK**.

i. Click **Save**.

5.5.2 Creating Separate Accounts for the Same User and Same Resource on a Single Target System

Two distinct accounts can be created for the same user and same resource on a single target system via access policy. For example, it is required to create two accounts, a user account and service account on a single AD instance. The Active Directory target system is represented by the AD User resource in Oracle Identity Manager. This is implemented in the following way:

1. Create a AD User resource.
2. Create the user, such as JohnD.
3. In the process form, mark UD_ADUSER_ORGNAME as the discriminator field so that two distinct accounts have different login IDs.
4. Create two access policies as follows:
   - **For regular account:**

     Access policy name: AP1

     Associated to role: Role1

     Resource to provision: AD User

     Process form having Discriminator field: User ID (UD_ADUSER_ORGNAME)

     Default value in access policy: Account1
   - **For service account:**

     Access policy name: AP2

     Associated to role: Role2

     Resource to provision: AD User

     Process form having Discriminator field: User ID (UD_ADUSER_ORGNAME)

     Default value in access policy: Account2

     Note: You must create a prepopulate adapter associated with the process form to generate the values for User ID so that unique values are generated for this field.
5. Assign Role1 and Role2 to JohnD. Note that you will not see the resource provisioned right after completion of role assignment. You must either wait for the Evaluate User Policies scheduled task to run automatically or you run this scheduled task manually.

   When Role1 is assigned to JohnD, the Account1 account is created in the AD User target system via the AP1 access policy. When Role2 is assigned to JohnD, Account2 is created in AD User via AP2. Therefore, two distinct accounts can be

created for the same user and same resource on a single target system via access policy.

5.5.3 Provisioning Multiple Instances of a Resource to Multiple Target Systems

The following are the broad-level steps to provision multiple instances of a resource object to multiple target systems via access policy:

1. Create an IT resource type by using the IT Resources Type Definition Form in the Oracle Identity Manager Design Console. For information about using this form, see "IT Resources Type Definition Form" in *Developing and Customizing Applications for Oracle Identity Manager*.
2. Create multiple IT resource instances of the IT resource type that you created in step 1. For information about creating IT resources, see "Creating IT Resources" on page 8-1.

   Here, IT resource instance is the account discriminator. See "Provisioning Multiple Instances of the Same Resource via Access Policy by Using Account Discriminator" on page 5-7 for information about account discriminator.

   Note: Display the process form default for ITResource. It is mandatory to display it. By doing so, you can successfully provision an application instance via access policy.

3. Create a process form with a field of type that you created in step 1.
4. Create a resource object.
5. Create a process definition, and associate the resource object and process form. For information about creating a process definition, see "Creating a Process Definition" in *Developing and Customizing Applications for Oracle Identity Manager*.
6. Create access policies associating a role and resource object. See "Creating Access Policies" on page 5-8 for details.

When you have two instances of the same resource on different physical server, you can use access policy to provision both the instances of a resource to the same user, JohnD. This is described with the help of the following scenario:

You have tow AD instances, one hosted on server with IP as 10.151.14.82 and another hosted on server with IP 130.35.66.254. The user is to be provisioned to both the instances via access policy-based provisioning. To achieve this:

1. Create a AD User resource.
2. Create an IT resource with name ADServer1 that represents the server with IP address as 10.151.14.82.
3. Create an IT resource with name ADServer2 that represents the server with IP address as 130.35.66.254.
4. Mark the AD Server (UD_ADUSER_SERVER) process form field as the discriminator field.
5. Create two access policies as follows:
   - **For the account to be created on ADServer1:**

     Access policy name: AP3

     Associated to role: Role3

Resource to provision: AD User

Process form having Discriminator field: AD Server (UD_ADUSER_SERVER)

Default value for ITResourceLookup field: ADServer1

- **For the account to be created on ADServer2:**

  Access policy name: AP4

  Associated to role: Role4

  Resource to provision: AD User

  Process form having Discriminator field: AD Server (UD_ADUSER_SERVER)

  Default value for ITResourceLookup field: ADServer2

6. Assign Role3 and Role4 to the user JohnD.

   When Role3 is assigned to JohnD, the account is created in the target system on ADServer1 via the AP3 access policy. When Role4 is assigned to JohnD, the account is created in the target system on ADServer2 via the AP4 access policy. Therefore, two distinct accounts are created for the same user and same resource on two different instances of the target system via access policy.

5.5.4 Limitation of Provisioning Multiple Instances of a Resource via Access Policy

Provisioning multiple instances of a resource via access policy has the following limitations:

- A single access policy cannot provision multiple instances of a resource to a user. Multiple access policies must be created to provision multiple instances of resource. You must create the same number of access policies as that of instances of same resource that is to be provisioned.
- If a resource object has a process form that has fields marked as account discriminator fields, then the value of these fields must be specified in any access policy that provisions that resource. Otherwise, issues might be encountered, such as multiple accounts might be provisioned when the policies are evaluated next time.
- If a resource object has a process form that has fields marked as account discriminator fields and if you use the access policy engine to provision this resource to one or more users, then the values of the account discriminator fields must remain constant throughout the lifecycle of the account. In other words, the values of the account discriminator fields must not be changed. This is because the access policy engine uses the resource object key and the account discriminator values to decide whether or not to provision a new account to the user.

  By modifying account discriminator values, you modify the basis on which the provisioning decision had been taken. and the behavior of the access policy engine cannot be determined. Therefore, it is recommended that you do not modify account discriminator values. And the process form values of the account discriminator fields must not be changed.
- If access policies are configured with different account discriminator values, they provision different accounts to the user.

Note: Account discriminator values that are different only in casing (for example, abc and aBc) are also treated as different values. With this data, two accounts are provisioned to the end user.

5.6 Troubleshooting Issues with Evaluate User Policy Scheduled Job

In some cases, the Evaluate User Policy scheduled task may not trigger, process users, or process only a few users. Any of the following could be the reasons:

- The Evaluate User Policy scheduled task ran, but there are no users marked for policy evaluation.
- Users were marked for policy evaluation, however the users are not active.
- Policy evaluation was done, however provisioning operations failed.

  In this case, the event handlers related to provisioning is in CANCELLED state. Therefore, no accounts or entitlements are provisioned to the users.
- The Evaluate User Policy scheduled task is not triggered due to an issue with the scheduler.

  In this case, the scheduler issue needs to be troubleshooted separately.

To identify if the issue is with access policy, provisioning, or scheduler, perform the steps mentioned in the MetaLink note 1563379.1.

Part III

Form Management

This part describes how to manage forms in Oracle Identity Manager by using the For Designer.

It contains the following chapter:

- Chapter 6, "Managing Forms"

6
Managing Forms

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

This section contains the following topics:

- Creating Forms By Using the Form Designer
- Searching Forms By Using the Form Designer
- Modifying Forms By Using the Form Designer

> **Note:** Before you start performing the procedures described in this section, it is recommended that you review the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

6.1 Creating Forms By Using the Form Designer

To create forms by using the Form Designer:

1. Login to Oracle Identity System Administration.
2. Create and activate a sandbox. A warning message is displayed if no sandbox is activated. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
3. In the left pane, under Provisioning Configuration, click **Form Designer**. The Form Designer page is displayed.
4. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Form page is displayed.
5. In the Resource Type field, specify a resource object with which you want to associate the form. To do so:
   a. Click the lookup icon next to the Name field. The Search and Select: Name dialog box is displayed.
   b. In the Name field, enter the name of the resource object you want to search. You can leave this field blank if you want to display all resource objects.
   c. Click **Search**. The resource objects that match the search condition are displayed.

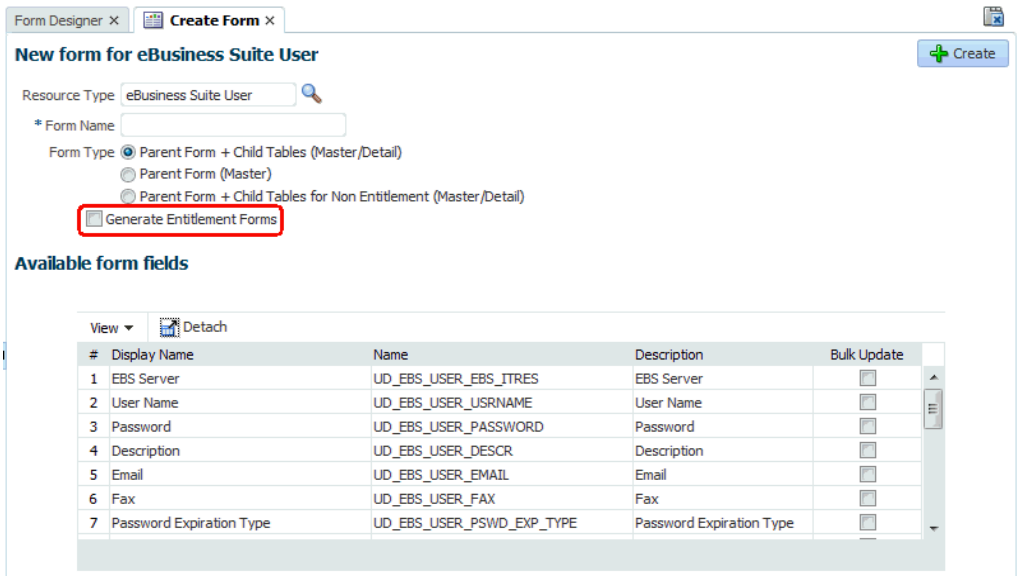
d. Select the resource object that you want to associate with the form, and click **OK.** The resource object name is displayed in the Name field of the Create Form page.

6. In the Form Name field, enter a form name.

7. In the Available form fields section, a list of form field names along with description and Display Name are displayed. These fields are available for the form you are creating. For each available form field, you can select the Bulk Update option. Selecting this option makes the form field available for updating the entities in bulk.

8. (Optional) By default, the **Parent Form + Child Tables (Master/Detail)** option is selected. You can select a different **Form Type** option.

9. (Optional) Select the **Generate Entitlement Forms** option if you want to associate the new form with the entitlements. Using this form, users can provide additional information that might help an approver during the approval process. The following is a sample screenshot:



Note: The **Generate Entitlement Forms** option is displayed only for complex entitlements. A complex entitlement is represented by child object having at least two attributes, one of them marked as Entitlement attribute.

10. Click **Create**. A message is displayed stating that the form is created.

11. If required, you can export the sandbox to store all the changes made in your sandbox. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

12. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

6.2 Searching Forms By Using the Form Designer

To search forms by using the Form Designer:

1. In Oracle Identity System Administration, under Provisioning Configuration, click **Form Designer**. The Form Designer page is displayed.
2. From the Resource Type list, select the type of resource object associated with the form.
3. Click **Search**. The forms that match your search condition are displayed. For each form, the search result displays the form name, form type, and resource type.

6.3 Modifying Forms By Using the Form Designer

To modify a form by using the Form Designer:

See Also: Section 6.3.1, "Removing or Hiding Form Attributes"

1. Create and activate a sandbox. A warning message is displayed if no sandbox is activated. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
2. In the Form Designer page, search for the form you want to modify.
3. In the Search Results table, select the form you want to modify.
4. From the Actions menu, select **Open**. Alternatively, click **Open** on the toolbar. Otherwise, click the form name in the search results table.

   The Manage Form page displays the form attributes in the Object Information section. The Standard and Custom sections list the standard and custom fields of the form. You can edit the standard fields, and create and edit custom fields in these sections.
5. (Optional) If you want to associate a form with the entitlements, then you can regenerate the form to allow users to provide additional information that might help the approver during the approval process. To do so, click **Regenerate View**. In the Regenerate View popup window, select the **Generate Entitlement Forms** checkbox, as shown in the following sample screenshot.

Note:

- If you have upgraded Oracle Identity Manager to release 11.1.2.2.0, then you must regenerate all the forms to use this feature.
- The Generate Entitlement Forms option is displayed only for complex entitlements. A complex entitlement is represented by child object having at least two attributes, one of them marked as Entitlement attribute.

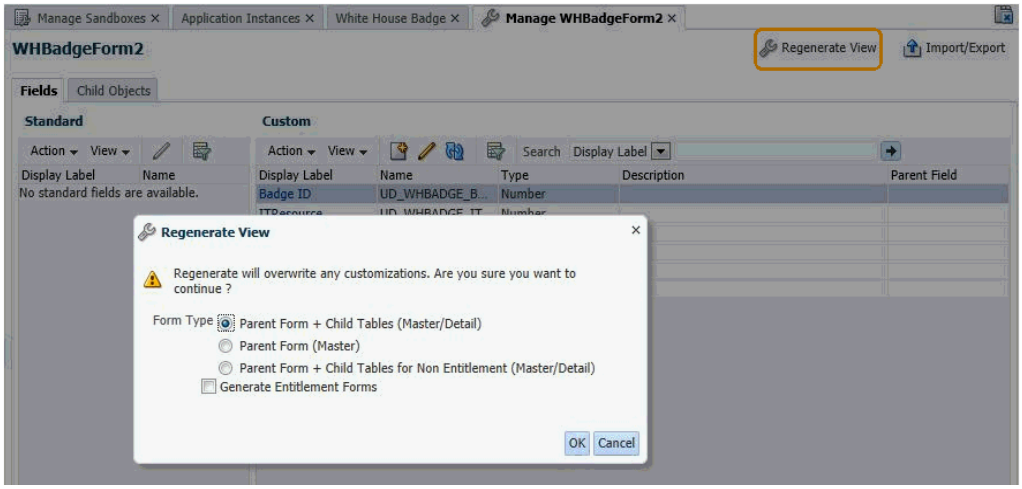


Table 6–1 lists the options in the Regenerate View popup window.

Table 6–1 Options in the Regenerate View Window

Option	Description
Parent Form + Child Tables (Master/Detail)	Selecting this option generates the appropriate account form. The account form includes all multi-valued attributes irrespective of whether the forms represent an entitlement or not.
Parent Form (Master)	Selecting this option generates the appropriate account form. The account form does not include any multi-valued attributes.
Parent Form + Child Tables for Non Entitlement (Master/Detail)	Selecting this option generates the appropriate account form. The account form includes all multi-valued attributes that do not represent an entitlement.
Generate Entitlement Forms	Selecting this checkbox generates the appropriate Entitlement forms. The entitlement form is generated only if the multi-valued attribute that represent an entitlement is complex. If the multi-valued attribute that represent an entitlement is scalar, then no form is generated.

6. If required, you can export the sandbox to store all the changes made in your sandbox. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
7. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

> **See Also:** "Configuring Custom Attributes" on page 7-1 for information about creating and modifying custom fields or user-defined fields (UDFs)

6.3.1 Removing or Hiding Form Attributes

To remove or hide a form attribute in Oracle Identity Self Service:

1. Log in to Oracle Identity Self Service.

2. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
3. Open the request catalog.
4. Search for and select the application instance whose resource form page must be updated, and the click **Add to Cart.**
5. Click **Checkout.**
6. On the Cart Details page, under the Details section, the application instance form and its attributes are displayed.
7. Click **Customize** to open WebCenter Composer. The page opens in customization mode.
8. Click **Structure**. The object tree is displayed.
9. If you want to delete a form attribute, select the UI component and click the delete icon in the Composer panel at the top of the page.

   If you want to hide a form attribute, click **Edit.** Then, select the UI component and set the Visible property to `false`.
10. Click **Close** to leave customization mode.
11. If required, you can export the sandbox to move the change from the test to production environment. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Developing and Customizing Applications for Oracle Identity Manager.*
12. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Developing and Customizing Applications for Oracle Identity Manager.*

Part IV

System Entities

This part describes how to extend users, roles, organizations, and request catalogs.

It contains the following chapter:

- Chapter 7, "Configuring Custom Attributes"

7

Configuring Custom Attributes

Entity attributes are properties of the entity. The information about the user entity is stored in the form of attributes, such as first name, last name, user login, and password. There are default user attributes in Oracle Identity Manager. However, you can create custom user attributes by using the User form under System Entities in the Oracle Identity System Administration. The custom attributes are referred to as user defined fields (UDFs). Oracle Identity Manager lets you create UDFs for the user, role, resource, organization, and catalog entities.

This chapter describes how to create and manage UDFs in the following sections:

- Creating a Custom Attribute
- Creating a Custom Child Form
- Creating a Custom Child Form Attribute
- Modifying a Custom Attribute
- Adding a Custom Attribute
- Adding a Custom Attribute to an Application Instance Form
- Moving UDFs from Test to Production
- Synchronizing User-Defined Fields Between Oracle Identity Manager and LDAP
- Creating Cascaded LOVs
- Localizing Display Labels of UDFs
- Configuring a Field as Mandatory Attribute in the Request Catalog

---

Note: Before you start performing the procedures described in this section, it is recommended that you review the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

7.1 Creating a Custom Attribute

The searchable property controls whether or not the attribute can be used to perform searches. For user defined attributes, setting this property will result in the attribute being shown in the Search form. Default attributes do not support this property.

To create a custom attribute or UDF:

Note:

- Do *not* use ParentAccountId as a form field name. ParentAccountId is used to store system information.
- Do not define a UDF as User Status. This column name is reserved for internal use.

1. Log in to Oracle Identity System Administration.
2. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
3. To create a UDF for the user, organization, role, and catalog entities, click the component under System Entities on the left navigation pane of Identity System Administration.

   Catalog UDFs will not be available under Role VO. When ever a catalog UDF is added and customized in access request page, then the new UDF is available automatically in Role page.
4. In the Custom section of the Fields tab, click the **Create** icon. The Select Field Type dialog box is displayed.
5. Select a field type you want to create. The available field types are:
   - **Text:** Select this option to create a text field.
   - **Number:** Select this option to create a numeric field.
   - **Checkbox:** Select this option to create a checkbox field.
   - **Date:** Select this option to create a date type field.
   - **Lookup:** Select this option to create a lookup field in which users can search and select the value. Note that there are two types of lookups that you can create:
     - A drop-down list from which you can select a value.
     - A searchable picklist (ADF name input list of value), from which users can search and select the value. If you want to create a searchable picklist, then on the Create Lookup Field page, under the Advanced section, select **Searchable Picklist.**

Note: After you create a UDF for dependent lookups (a lookup field that is created with the **Constrain list by parent field value selection option** selected), you must set the partialTriggers property through WebCenter composer to refresh the values in the dependent lookup. To do so, see the procedure described in "Creating Cascaded LOVs" on page 7-21.

If you create a UDF in the User Details page, then the UDF is recommended to be in read-only mode. If the UDF is of drop-down or checkbox type, then you must customize it to read-only mode explicitly. To do so:

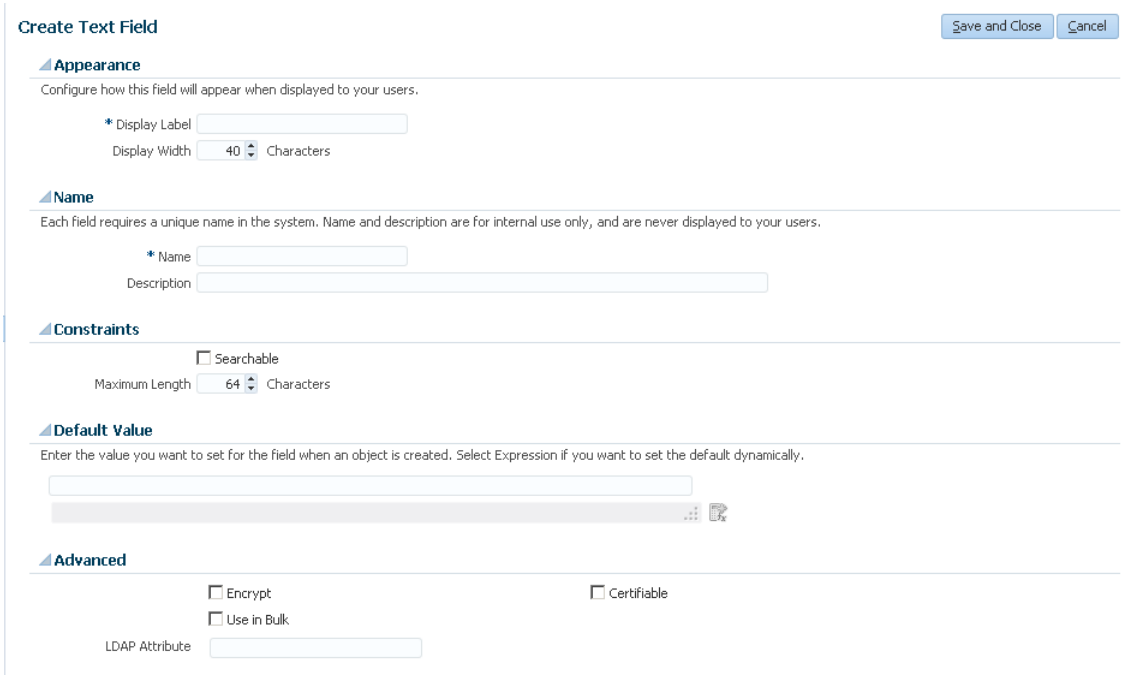
1. In the User Details page, click **Customize** to open WebCenter Composer. The page opens in customization mode.
2. Click the drop-down or checkbox region to edit its properties. In the pop-up window, click **Edit.**
3. In the Component Properties window, select the **Read Only** checkbox and click **OK.**
4. Click **Close** to close the page in customization mode.

Do not add drop-down UDF as outputText to a page if the value of the Meaning field has to be displayed.

6. Click **OK**. The page to create a custom field is displayed.

   As an example, Figure 7–1 shows the Create Text Field page. The rest of the procedure in this section has been based on creating a custom text field.

Figure 7–1 The Create Text Field Page



7. Enter values in the fields of the Create Text Field page. Table 7–1 lists the fields in the Create Text Field page. Depending on the type of field that you are creating, the fields on the Create Text Field page varies.

Table 7–1 Fields in the Create Text Field Page

Section	Field	Description
Appearance	Display Label	The custom field label that is displayed in the form. **Note:** Display Labels for forms designed by using the Form Designer must be specified in single default language, for example English. If there is a requirement to enter the Display Label in any other language, then the ROOT resource bundle (/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf) containing the Display Labels specified in the Form Designer can be translated to other languages. The translated labels is displayed when the form is displayed in the request catalog, Inbox, track requests, and other pages.
	Display Width	The display width in characters. If you do not specify a value for this field, then the length of the field is taken as default.
Name	Name	The unique custom field name. This field is of internal use only, and the value of this field is not displayed to the user.
	Description	The description of the custom field. This field is of internal use only, and the value of this field is not displayed to the user.
Constraints	Searchable	The searchable property controls whether or not the attribute can be used to perform searches. For user defined attributes, setting this property will result in the attribute being shown in the Search form. Default attributes do not support this property. **Note:** If you select the **Searchable** checkbox, then in the Advanced section, you cannot select **Encrypt.** A custom field that is marked as searchable cannot be encrypted.
	Maximum Length	The maximum length of the field in characters. **Note:** You can increase the maximum length for default and custom attributes by using the User form. However, decreasing the maximum length is not supported.
Default Value	Text field	The default value of the custom field. The value you specify in this field is set for the field when the object is created. **Note:** The field below the text field is grayed out and is not used.
Advanced	Encrypt	Determines whether the custom field must be encrypted. **Note:** If you select the **Encrypt** checkbox, then in the Constrains section, you cannot select **Searchable.** A custom field that is encrypted cannot be searchable.
	Use in Bulk	Determines whether the attribute is available in bulk operations.

Table 7–1 (Cont.) Fields in the Create Text Field Page

Section	Field	Description
	LDAP Attribute	Name of the attribute in the LDAP repository to which this custom attribute must map to. **Note:** Unless LDAP synchronization is enabled, setting a value for this field has no effect. For more information about enabling LDAP synchronization, see the "Configuring OIM Server" chapter in *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management*.
	Certifiable	Determines whether the attribute is certifiable. A requestable entity is available for certification only if it is marked as certifiable.

8. Click **Save and Close**. The UDF is created in the backend and is displayed in the Custom section of the Form Details page.
9. It is recommended that you export the sandbox to store all the changes made in your sandbox. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
10. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

When you create a UDF by using the Form Designer, it is created in the back-end, and is not available for use. To make it available for use to the user, you must include the UDF in the Oracle Identity Self Service page on which it is displayed. For information about including a UDF in the Oracle Identity Self Service page, see "Adding a Custom Attribute" on page 7-9.

7.2 Creating a Custom Child Form

Application instance forms can have child forms. Note that at some places in this guide, the term **resource form** has been used to refer to **application instance forms.**

To create a custom child form:

1. Log in to Oracle Identity System Administration.
2. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

> **Note:** You must ensure that sandbox in which the application instance form for which you are creating the child form must be published. If it is not published, then you must perform the procedure described in this section in the same sandbox in which the application instance form was created.

3. In the left pane, under Provisioning Configuration, click **Form Designer**. The Form Designer page is displayed.
4. Search for and open the application instance (resource) form for which you want to create a child form as follows:

a. Specify a value for the Resource Type lookup field.

b. Click **Search.**

   A list of all resource forms (application instance forms) that meet the search criteria is displayed.

c. From this list, select the form to open. Alternatively, click **Open** on the toolbar.

   The Manage *APP_INSTANCE_FORM_NAME* page is displayed.

5. On the Child Objects tab, click the **Add** icon on the toolbar. The Add dialog box is displayed.
6. In the Name field, enter the name of the child form. In the Description field, enter a description of the child form. Then, click **OK.** The child form is created in the backend and is displayed in the Child Objects tab of the application instance form for which it was created.

   For information about adding a new child form attribute, see "Creating a Custom Child Form Attribute" on page 7-6.
7. Click **Regenerate View** to regenerate the application instance form associated with the child form. If you do not regenerate the view the child form will not be available in the page for use on which you want it to be displayed.
8. It is recommended that you export the sandbox to store all the changes made in your sandbox. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
9. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

7.3 Creating a Custom Child Form Attribute

To create a custom child form attribute:

Note: Do *not* use ParentAccountId as a form field name. ParentAccountId is used to store system information.

---

1. Log in to Oracle Identity System Administration.
2. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

Note: You must ensure that sandbox in which the child form for which you are creating the attribute must be published. If it is not published, then you must perform the procedure described in this section in the same sandbox in which the child form was created.

---

3. In the left pane, under Provisioning Configuration, click **Form Designer**. The Form Designer page is displayed.
4. Search for and open the parent form (application instance form) of the child form in which you want to create an attribute. See Step 4 of "Creating a Custom Child

Form" on page 7-5 for information about searching and opening a form.

The Manage *APP_INSTANCE_FORM_NAME* page is displayed.

5. On the Child Objects tab, from the list of child forms, select the child form in which you want to create the attribute. The Manage *CHILD_FORM_NAME* page is displayed.
6. In the Custom section of the Fields tab, click the **Create** icon. The Select Field Type dialog box is displayed.
7. Select a field type you want to create. The available field types are:
   - **Text:** Select this option to create a text field.
   - **Number:** Select this option to create a numeric field.
   - **Checkbox:** Select this option to create a checkbox field.
   - **Date:** Select this option to create a date type field.
   - **Lookup:** Select this option to create a lookup field in which users can search and select the value.
8. Click **OK**. The page to create a custom field is displayed.

   The rest of the procedure in this section has been based on creating a custom lookup field.
9. Enter values in the fields of the Create Lookup Field page. Table 7–2 lists the fields in the Create Lookup Field page:

Table 7–2 Fields in the Create Lookup Field Page

Section	Field	Description
Appearance	Display Label	The custom field label that is displayed in the form. **Note:** Display Labels for forms designed by using the Form Designer must be specified in single default language, for example English. If there is a requirement to enter the Display Label in any other language, then the ROOT resource bundle (/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf) containing the Display Labels specified in the Form Designer can be translated to other languages. The translated labels is displayed when the form is displayed in the request catalog, Inbox, track requests, and other pages.
	Display Width	This attribute will specify the width for LOV UDF on the screen. **Note:** When creating a Lookup type UDF, the recommended value of Display Width is `40`.
	Help Text	Field-level help text that is displayed to the users as a tooltip.
Name	Name	The unique custom field name. This field is of internal use only, and the value of this field is not displayed to the user.
	Description	The description of the custom field. This field is of internal use only, and the value of this field is not displayed to the user.
Constraints	Searchable	Determines if the custom field can be searched by the user.

Table 7–2 (Cont.) Fields in the Create Lookup Field Page

Section	Field	Description
	Maximum Length	Determines the maximum length of the value that can be provided.
List of Values	Lookup Type	The lookup whose values are displayed to the user as a list of available values. You can either specify an existing lookup type or create a new one. **Note:** If you are creating a new lookup, then the name of this new lookup must not be the same as that of the UDF (of type lookup) that you are creating. Otherwise, the lookup is not displayed in the Manager User page.
Default Value	Drop-down list	The default value of the custom field. The value you specify in this field is set for the field when the object is created. **Note:** The field below the down-down list is grayed out and is not used.
Advanced	Entitlement	Determines whether the custom field is an entitlement. **Note:** If you are creating a child form with a lookup field for entitlement (in other words, the Entitlement field is selected), then you must select **Searchable** and **Searchable Picklist** options too.
	Use in Bulk	Determines whether the attribute is available in bulk operations.
	Searchable Picklist	Determines whether the custom field is a input list of values. This is applicable to Lookup field.

10. Click **Save and Close**. The UDF is created in the backend and is displayed in the Custom section of the Form details page.
11. Click **Re-generate View**.
12. It is recommended that you export the sandbox to store all the changes made in your sandbox. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
13. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

7.4 Modifying a Custom Attribute

To modify a custom attribute that you created for a form:

1. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
2. In the Form Designer, search and open the form which contains the custom attribute you want to modify.
3. In the Custom section, select the custom attribute that you want to modify.
4. Click the **Edit** icon on the toolbar. Alternatively, click the Display Name of the attribute. The page to edit the field is displayed.

5. Modify the values in the fields by referring to Table 7–1. Note that all the fields listed in Table 7–1 are editable.
6. Click **Save and Close**.
7. Click **Re-generate View**.
8. It is recommended that you export the sandbox to store all the changes made in your sandbox. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
9. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

7.5 Adding a Custom Attribute

When you create a UDF, it is created only in the backend, and is not available in the page for use on which you want it to be displayed.

Note:

- Adding a custom attribute is always in relation to one of the following entities: User, Organization, Role, or Catalog.
- When catalog UDFs are customized to show in the first page of the Create Role wizard, they are also shown in the summary page of the wizard. But when role UDFs are customized to show in first page of the Create Role wizard, they are not shown in the summary page of the wizard. The summary page must be separately customized for these role UDFs to be displayed.

To display a UDF in a page in Oracle Identity Self Service:

1. Create the UDF by using the User form under System Entities in Identity System Administration. For example, you can create a UDF for the Create User page.

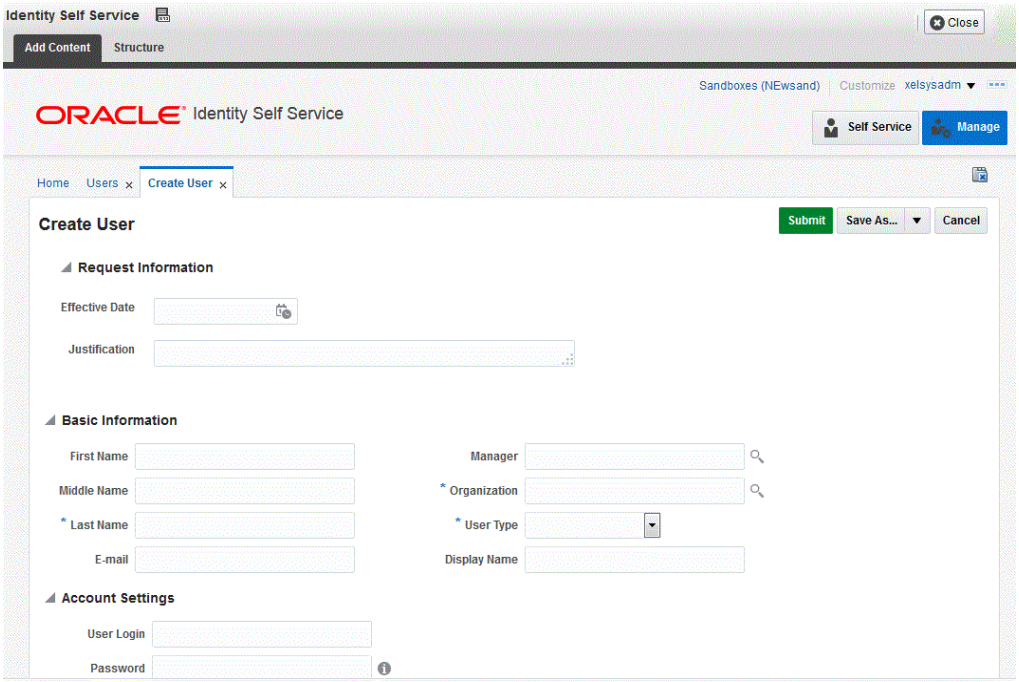
   See "Creating a Custom Attribute" on page 7-1 for information about creating a UDF.

   Note: After adding a UDF through the User form, logout of both Oracle Identity System Administration and Oracle Identity Self Service, and then login again to be able to see the newly added UDF and use it for customization.

2. Log in to Oracle Identity Self Service as the system administrator.
3. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
4. Click **Manage**. The Home tab displays the different Manage option. Click **Users**. The Manage Users page is displayed.
5. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar. The Create User page is displayed with input fields for user profile attributes.

6. Click **Customize** at the upper right corner of the page to open WebCenter Composer. The Create User page opens in customization mode as shown in Figure 7–2.

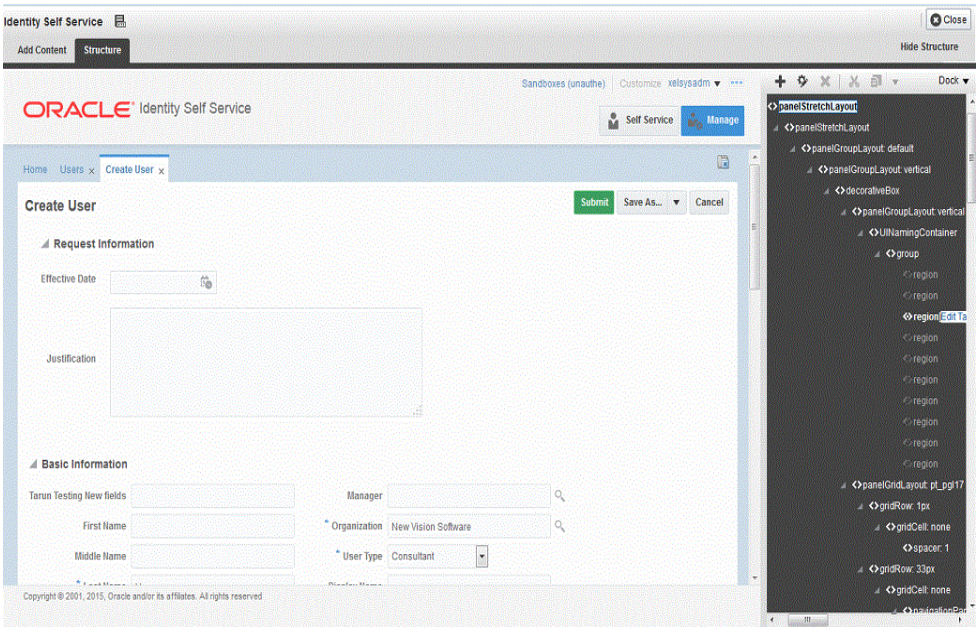
Figure 7–2 Create User Page in Customization Mode



7. Enter values for all mandatory fields.

8. Select **Structure** tab. The object tree is displayed as shown in Figure 7–3.

Figure 7–3 Object Tree Page in Customization Mode



9. Select the section of the page on which you want to add the UDF.

10. In the Confirm Task Flow Edit dialog box, click **Edit** to confirm the edit task. The corresponding ADF component in the object tree is selected.
11. Select the **panelFormLayout** component, and click the **Add** icon. The Add Content dialog box is displayed.
12. Depending on the entity or area on which the UDF was added, select the data component, and then the view object. Table 7–3 lists the entities, pages, data components, and view objects that must be selected.

Note: Adding VO as tables is not supported.

Table 7–3 Entities and Corresponding Data Components and View Objects

Entity	Page	Data Component	View Object
User	Create User	Data Component - Catalog	userVO
	Modify User	Data Component - Catalog	userVO
	Search Users	Data Component - Manage Users	UserVO1
	View User Details	Data Component - Manage Users	UserVO1
	My Information	Data Component - My Information	UserVO1
	New User Registration	Data Component - User Registration	UserVO1
Role	Create Role	Data Component - Role	RoleDetailsVO
	Modify Role	Data Component - Role	RoleDetailsVO
	Search Roles	Data Component - Role	RoleVO1
Organization	Create Organization	Data Component - Organization	EditOrgVO
	Modify Organization	Data Component - Organization	EditOrgVO
	Search Organizations	Data Component - Organization	OrganizationVO
Catalog	Access Request	Data Component - Catalog	▪ Catalog results table: CartItemsVO1 ▪ Cart items under Edit Cart Popup: CartItemsVO ▪ Catalog details for a selected cart item either under catalog results table or edit cart popup: EditCartItemsVO
Certification	User Certification	Data Component - Certification	UserCertificationUserVO1
			UserCertificationUserEntitlementVO1
	Role Certification	Data Component - Certification	RoleCertificationRoleVO1
			RoleCertificationMemberVO1

Table 7–3 (Cont.) Entities and Corresponding Data Components and View Objects

Entity	Page	Data Component	View Object
			RoleCertificationPolicyVO1
	Application Instance Certification	Data Component - Certification	ApplicationCertificationApplicationVO
			ApplicationCertificationEntitlementVO
	Entitlement Certification	Data Component - Certification	EntitlementCertificationEntitlementVO
			EntitlementCertificationEntitlementMemberVO

13. Scroll to find the UDF that you added and click **Add.** If the UDF is not displayed, then refresh the content by clicking the **Refresh** icon at the top right hand corner of the dialog box.
14. Depending on the custom attribute that you created in Step 1 and the type of UDF that you want to display, select one of the following items from the menu:

    For a UDF of Text or Number type:

    - ADF Output Text
    - ADF Output Text w/Label
    - ADF Output Formatted
    - ADF Output Formatted w/Label
    - ADF Input Text
    - ADF Input Text w/Label
    - ADF Label
    - ADF Readonly Input Text w/Label
    - ADF Table Column

    For a UDF of Checkbox type:

    - ADF Select Boolean Checkbox
    - ADF Table Column

    For a UDF of Date type:

    - ADF Input Date w/Label
    - ADF Table Column

    For a UDF of Lookup type:

    - ADF Input List Of Value (select only for searchable PickList)
    - ADF Select One Choice (select only for non-searchable PickList; this option is not visible for a searchable PickList for which you must select ADF Input List of Value)
    - ADF Table Column (select when adding a column within an af:table)

    For example, if you have created a UDF of Text type, then select **ADF Input Text w/Label.** Similarly, if you created a searchable UDF of Lookup type, then select **ADF Input List of Value.** As an example, Figure 7–4 shows options for a UDF of

Text type.

Figure 7–4 Options for Adding a UDF of Text Type

15. Click **Close** to close the Add Content dialog box.

Note: If two attribute labels are displayed for the same field, then add the attribute that does not end with __C.

---

16. From the object tree on the Editing Page, select the UDF on the page, and click the **Show properties** icon. The Component Properties page is displayed.
17. On the Display Options tab:
    a. Select **Auto Submit.**
    b. If you have added the UDF on the user form, then in the Value Change Listener field, enter `#{pageFlowScope.cartDetailStateBean.attributeValueChangedListener}`.

       If you have added the UDF on a form other than the user form, then copy the value of the Value Change Listener field from any of the existing fields on the form and paste it as the value of the Value Change Listener field for the newly added UDF.
    c. If you want to mark this attribute as mandatory, then change the **Required** and **Show Required** properties to `true`. To set the Show Required property, select the **Show Required** option. In the Required field, select **Expression Editor**, and in the Expression Editor field, enter the value as `true`.
    d. If you want to display this attribute as read-only, then select the checkbox for the **Read Only** property.
    e. If you want to bind this attribute to a custom-managed bean method, then change the **Value** property.

       The custom-managed bean method must include a call to the original method binding. For more information, see "Developing Managed Beans and Task Flows" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
18. Click **OK.**
19. Click **Close** to leave customization mode.
20. It is recommended that you export the sandbox, in case if you intend to move the change from test to production environment. See "Managing Sandboxes" in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for detailed instructions on exporting a sandbox.
21. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

To remove a UDF, you can use the customization mode to open the WebCenter Composer. In the customization mode, select the component or UDF that you want to remove, and then delete it or set the rendered property on that UDF to false.

7.5.1 Enabling the Submit Button After Adding a UDF to the Modify User Form

After adding a new UDF to the modify user form by customizing the UI using Web Composer, the Submit button of the form is not enabled when you try to modify a user. But modification of other user form fields enable the Submit button.

To avoid this issue, when you add a new UDF to the modify user form for the first time:

1. Create a sandbox and activate it. Open the page that contains the UDF, and click **Customize**.
2. Select **Structure**.

3. Note the value of the valueChangeListener property of a predefined or default field. To do so:
   a. Click the predefined field, and then click **Edit** to open the Component Properties dialog box.
   b. Copy the value of the valueChangeListener property.
4. Add the new UDF to the form, as described in "Adding a Custom Attribute" on page 7-9.
5. Export the sandbox as a ZIP file.
6. Delete the sandbox without publishing it.
7. Extract the ZIP file, and edit the jsff.xml file for the specific screen.
8. Add the following attributes to the ADF tag, for example af:inputText, for the UDFD field, as shown:

```
valueChangeListener=VALUE_COPIED_IN_STEP3
autoSubmit="true"
```

   The resulting XML will look similar to the following:

```
<?xml version='1.0' encoding='UTF-8'?>
<mds:customization version="11.1.1.61.92"
xmlns:mds="http://xmlns.oracle.com/mds" motype_local_name="root"
motype_nsuri="http://java.sun.com/JSP/Page">
   <mds:move node="_xg_12" parent="_xg_pfl5" position="last"/>
   <mds:insert parent="_xg_pfl5" position="last">
      <af:inputText xmlns:af="http://xmlns.oracle.com/adf/faces/rich"
value="#{bindings.JobCode__c.inputValue}"
label="#{bindings.JobCode__c.hints.label}"
required="#{bindings.JobCode__c.hints.mandatory}"
columns="#{bindings.JobCode__c.hints.displayWidth}"
maximumLength="#{bindings.JobCode__c.hints.precision}"
shortDesc="#{bindings.JobCode__c.hints.tooltip}" id="dtrt_dc_628826708"
autoSubmit="true"
valueChangeListener="#{pageFlowScope.cartDetailStateBean.attributeValueChanged
Listener}">
     <f:validator xmlns:f="http://java.sun.com/jsf/core"
binding="#{bindings.JobCode__c.validator}"/>
      </af:inputText>
   </mds:insert>
   <mds:move node="_xg_19" parent="_xg_pfl5" position="last"/>
   <mds:move node="_xg_20" parent="_xg_pfl5" position="last"/>
   <mds:move node="_xg_27" parent="_xg_pfl5" position="last"/>
   <mds:move node="_xg_23" parent="_xg_pfl5" position="last"/>
   <mds:move node="_xg_41" parent="_xg_pfl5" position="last"/>
</mds:customization>
```

9. Create the ZIP file for the sandbox.
10. Import the sandbox.
11. Publish the sandbox.

7.5.2 Adding a Custom Attribute Category into Create User Form

You must customize the Create User or Modify User form to add a new category of fields. To do so:

1. Log in to Oracle Identity Self Service.

2. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
3. Click **Manage**. The Home tab displays the different Manage option. Click **Users**. The Manage Users page is displayed.
4. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar. The Create User page is displayed with input fields for user profile attributes.
5. Click **Customize** at the upper right corner of the page to open WebCenter Composer. The Create User page opens in customization mode.
6. Enter values for all mandatory fields.
7. Select **Structure** tab. The object tree is displayed.
8. Click the first field of the Create User form and select its ancestor **panelGroupLayout** component.
9. Click the **Add Content** icon.
10. In the Add Content dialog box, click **Web Components.**
11. Click **Add** next to the **ShowDetailHeader** component.
12. Click **Close.**
13. Select the newly added **ShowDetailHeader** component and click **Edit** to open the Component Properties dialog box.
14. Modify the value of Size to `2`.
15. Modify the default value of Text with a suitable value.
16. Click **Apply** and **Close.**
17. Click the **Add Content** icon.
18. In the Add Content dialog box, click **Web Components**, if not already open.
19. Click **Add** next to the **PanelFormLayout** component.
20. Click **Close.**
21. Add fields into this new panelFormLayout component as described in Step 11 in Adding a Custom Attribute.
22. Click **Close** to leave customization mode.
23. It is recommended that you export the sandbox, in case if you intend to move the change from test to production environment. See "Managing Sandboxes" in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for detailed instructions on exporting a sandbox.
24. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

7.5.3 Customizing Unauthenticated Page

You can customize an unauthenticated page for example New User Login or Self Registration page. To do so:

1. Log in to Oracle Identity Self Service.

2. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
3. Click **Self Service**. The Home tab displays the different Self Service option.
4. Click **Customize** at the upper right corner of the page to open WebCenter Composer. The Home page opens in customization mode.
5. Select **Structure** tab. The object tree is displayed.
6. Select the area on the screen where all other tiles like My Information, My Access and so on are present.
7. In the right hand side panel, select the last *gridRow*, right click and select **Show Component**.

   Incase the Unauthenticated box does not immediately appear on the screen, you may have to close the screen and reopen.
8. Unauthenticated option gets added to the screen. This box has a drop-down list of all unauthenticated pages in Self Service Console. You can select any one screen that you would like to customize.

   For detailed steps on how to add a custom attribute see, Adding a Custom Attribute.

7.6 Adding a Custom Attribute to an Application Instance Form

When you create a custom attribute (UDF) on an application instance form, it is created only in the backend, and is not available in the page for use on which you want it to be displayed. The following are the options available to display the UDF in a page in Oracle Identity Self Service:

- Regenerating View
- Updating the Application Instance Form By Using WebCenter Composer

7.6.1 Regenerating View

To display the UDF in a page in Oracle Identity Self Service:

1. Log in to Oracle Identity System Administration.
2. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

   Note: You must ensure that sandbox in which the application instance form for which you are adding a custom child attribute must be published. If it is not published, then you must perform the procedure described in this section in the same sandbox in which the application instance form was created.
3. In the left pane, under Provisioning Configuration, click **Form Designer**. The Form Designer page is displayed.
4. Search for and open the application instance form whose child form (containing the UDFs that you added) must be displayed in a page in Oracle Identity Self Service.

5. On the Child Objects tab, click **Regenerate View.**

Note:

- The Regenerate View dialog box is displayed. Select the appropriate options for **Form Type** and **Generate Entitlement Forms**. See "Modifying Forms By Using the Form Designer" on page 6-3 for information about the Form Type and Generate Entitlement Forms options.
- Any customization made to the page is lost when you click **Regenerate View**.

6. It is recommended that you export the sandbox, in case if you intend to move the change from test to production environment. See "Managing Sandboxes" in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for detailed instructions on exporting a sandbox.
7. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

7.6.2 Updating the Application Instance Form By Using WebCenter Composer

To display the UDF in a page in Oracle Identity Self Service:

1. Create the UDF by using the Form Designer.
2. Log in to Oracle Identity Self Service.
3. Create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
4. In the left pane, under System Entities, click **Catalog.** The Catalog page is displayed.
5. Search for and select the application instance whose resource form page must be updated, and the click **Add to Cart.**
6. Click **Checkout.**
7. On the Cart Details page, under the Details section, the application instance form and its attributes are displayed.
8. Click **Customize** to open WebCenter Composer. The page opens in customization mode.
9. Enter values for all mandatory fields.
10. From the View menu at the upper left corner of the page, select **Structure.** The object tree is displayed.
11. Under the Details section, select and click the attributes of the application instance form. A message confirming whether you want to edit the page is displayed.
12. Click **Edit.** In the object tree, the ADF component corresponding to the selection made in the preceding step is selected.
13. Click **Add Content**. The Add Content dialog box is displayed.
14. Select the data component. To do so:

a. Select **Data Component - Catalog.**

b. Search for ***APP_INSTANCEVO*** and then click **Open.** Here, *APP_INSTANCE* is the name of the application instance for which the attributes are added.

15. Scroll to find the UDF that you added. If the UDF is not displayed, then refresh the page.
16. Select the UDF on the page, and click **Add.**
17. Click **Close** to leave customization mode.
18. It is recommended that you export the sandbox to move the change from the test to production environment. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
19. Publish the sandbox. For detailed instructions on publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

7.7 Moving UDFs from Test to Production

The following sections discuss the procedure to move a UDF added to entities from test to production:

- Moving UDFs Added to Entities
- Moving UDFs Added to Catalog Entities

See Also:

- "Limitations of the Test to Production procedures" on page 13-27 for information about test to production limitations.
- "Handling Concurrency Conflicts" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about handling concurrency conflicts when multiple users customize an application by using sandboxes and troubleshooting concurrency issues

7.7.1 Moving UDFs Added to Entities

Moving a UDF that is added to a User, Roles, Organization or Application Instance entity from test to production consists of the following steps:

- Exporting the UDF from the Test Environment
- Importing the UDF into the Production Environment

Note: Before you perform these procedures, ensure that you do not have any popup blockers enabled in your browser and that you have a supported Java Runtime Environment (JRE) installed in the browser. This is because the Deployment Manager uses a popup window and it requires JRE to be installed in the browser.

7.7.1.1 Exporting the UDF from the Test Environment

To export the UDF from the test environment:

1. Log in to Oracle Identity System Administration.

2. Under System Configuration, click **Export.**
3. Search for the desired metadata, User Metadata, Role Metadata, Organization Metadata, or Application instances. A list of all available metadata is displayed.
4. Select the UDF that you want to move from test to production, and then click **Select Children.**
5. Click **Select Dependencies,** and then click **Confirmation.**
6. Click **Add for Export.**
7. In the confirmation message that is displayed, click **OK** to exit the wizard.
8. Click **Export.** Alternatively, provide description and then click **Export.**
9. Specify the location to which the content must be exported. A message confirming that the export was successful is displayed.
10. Export the sandbox from the test environment to store all the changes made in your sandbox. For detailed instructions on exporting a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

Note:

- The sandbox exported here must be the same, which has been used while creating and adding custom UDFs.
- The sandbox must not have been published before exporting, because there is no way to export the published sandbox.

7.7.1.2 Importing the UDF into the Production Environment

To import UDF into the production environment:

1. In Oracle Identity Manager System Administration, under System Configuration, click **Import.**
2. Specify the path to the XML file that was exported from the test environment by using the Deployment Manager.
3. Click **Add File, Import,** and then confirm the import. A message confirming that the import was successful is displayed.
4. Import the sandbox exported from the test environment. For information about importing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
5. Activate the sandbox to verify the changes. For information about activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
6. Publish the sandbox after you verify the changes. For information about publishing a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*

7.7.2 Moving UDFs Added to Catalog Entities

The procedure to move a UDF added to a catalog entity from test to production is discussed later in this guide. See "Test to Production procedures for Catalog customizations" on page 13-25 for more information.

7.8 Synchronizing User-Defined Fields Between Oracle Identity Manager and LDAP

Note:

- LDAP synchronization can be enabled during or any time after installing Oracle Identity Manager. See "Enabling LDAP Synchronization in Oracle Identity Manager" in the *Oracle Fusion Middleware Integration Guide for Oracle Identity Management Suite* for information about enabling postinstallation LDAP synchronization in Oracle Identity Manager.
- While creating/modifying an attribute using Form Designer, provide a value against LDAP Attribute. This is the value of LDAP attribute name against which the user-defined field (UDF) is synchronized, and applicable only in LDAP sync enabled environment.
- If you are using an OUD LDAP directory, then the Oracle Identity Manager custom attribute name must not contain a space. OUD does not allow creating a custom attribute with space in the attribute name.

If you enable LDAP synchronization any time after creating one or more UDFs, then you must synchronize these UDFs with the corresponding LDAP attributes. To do so, by using the Form Designer, search for and open the form containing the UDF, and then save it (no need to make any other change). Repeat this process of opening the form containing the UDF and then saving it for all UDFs created before enabling LDAP synchronization.

7.9 Creating Cascaded LOVs

To create cascaded LOVs on the My Information page:

Note: In Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) or later, LOVs cannot be added on the Self-Registration Page.

1. Log in to Oracle Identity System Administration.
2. Create and activate a sandbox, for example `SUJ`. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developer's Guide for Oracle Identity Manager*.
3. Under System Entities in the left pane, click **User.**
4. Create the following UDFs of Lookup Type**:**
   - parent - `ParentChoice`
   - dependent - `DepChoice`

     While creating `DepChoice`, make it dependent on the UDF `ParentChoice`, and map the values. To do so:

     a. In the List of Values section, search for the parent field and select it.

Select **Constrain list by parent field value selection**. This enables the fields to set the parent dependency details.

b. Select the required **Parent Choice List** and set the **Value Map**.

5. Click **Save and Close**.
6. Export the sandbox.

   The sandbox is stored as sandbox_SUJ.zip.
7. Unzip the sandbox_SUJ.zip file, and perform the following steps:

   a. In the file \persdef\sessiondef\oracle\iam\ui\runtime\form\model\user\view\mdssys\cust\site\site\userVO.xml.xml, under tag <ViewAttribute Name="DepChoice__c", search for the following text:

   ```
   <Property Name="CascadingParentChoiceList"
   Value="ParentChoice__c"/>

   <Property Name="CascadingRelationshipId" Value="100000000002523"/>
   ```

   b. Copy the text in Step 7 a to \persdef\oracle\iam\ui\common\model\user\view\mdssys\cust\site\site\UserVO.xml.xml file under tag <ViewAttribute Name="DepChoice__c".

   c. In the file \persdef\sessiondef\oracle\iam\ui\runtime\form\model\user\view\mdssys\cust\site\site\userVO.xml.xml, search for the following text:

   ```
   </mds:insert>
   <mds:insert parent=" userVO " position="last">
   <ViewAccessor Name="LOVVA_For_DepChoice__c"
   ViewObjectName="oracle.adf.businesseditor.model.views.CascadingLookups "
   xmlns="http://xmlns.oracle.com/bc4j">

     <ParameterMap>
               <PIMap Variable="Bind_RelationshipId">
                  <TransientExpression Name="expression"
   access="local"><![CDATA[structureDef.findAttributeDef("DepChoice__c
   ").getProperty("CascadingRelationshipId")]]></TransientExpression>
               </PIMap>
               <PIMap Variable="Bind_ParentLookupCode">
                  <TransientExpression Name="expression"
   access="local"><![CDATA[ParentChoice __c]]></TransientExpression>
               </PIMap>
            </ParameterMap>

   </ViewAccessor>
   </mds:insert>
   </ParameterMap>
   </ViewAccessor>
   </mds:insert>
   ```

   d. In the file \tmp\persdef\oracle\iam\ui\common\model\user\view\mdssys\cust\site\site\UserVO.xml.xml search for the below text and replace it with the text copied in step 7 c . Change userVO to UserVO:

   ```
   </mds:insert>
   <mds:insert parent="UserVO" position="last">
   <ViewAccessor Name="LOVVA_For_DepChoice__c"
   ViewObjectName="oracle.adf.businesseditor.model.views.Lookups"
   ```

```
xmlns="http://xmlns.oracle.com/bc4j">
<ParameterMap>
<PIMap Variable="Bind_LookupType">
<TransientExpression><![CDATA['Lookup.Conditions.Severity']]></TransientEx
p
ression>
</PIMap>
</ParameterMap>
</ViewAccessor>
</mds:insert>
```

e. In the file \persdef\sessiondef\oracle\iam\ui\runtime\form\model\user\view\mdssys\cust\site\site\userVO.xml.xml, search for the following text:

```
<mds:insert parent="userVO" position="last">
      <Properties xmlns="http://xmlns.oracle.com/bc4j">
         <Property Name="__INTERNAL_EXPR_VALUE_OVERRIDES__"
Value="userEO"/>
      </Properties>
   </mds:insert>
```

f. Copy the text from 7 e to file persdef\oracle\iam\ui\common\model\user\view\mdssys\cust\site\site\UserVO.xml.xml and change userVO to UserVO and userEO to UserEO.

8. Recreate the zip file with same name as in Step 6.

   For example, `$zip -r sandbox_SUJ.zip*`

9. Delete the sandbox `SUJ` from Oracle Identity System Administration.
10. Import the modified sandbox_SUJ.zip created in Step 8.
11. Logout from Oracle Identity System Administration.
12. Log in to Oracle Identity Self Service.
13. Activate the sandbox, `SUJ`.
14. In the left pane, under **My Profile**, click **My Information**. The My Information page is displayed.
15. Click **Customize** to customize the My Information page while the sandbox is active in Oracle Identity Self Service.
16. Add parent UDF and child UDF (created in Step 4) on the page as **Select one choice** component.
17. Select ParentChoice and click **Edit Property** and copy the Id of parent component. Set the **auto submit property** to **true**.
18. Select DepChoice and click **Edit Property** and paste the id value of ParentChoice UDF copied in Step 17 to the partailTrigger field.
19. Publish the sandbox.

Note: For any LOV, the user details page displays the lookup code as the output text value. To display the LOV lookup value on the user details page, create a searchable picklist (ADF name input list of value), and then make it read-only.

Specifying Cascaded LOVs Without NULL Value

When you set the value of the required property to true in the attributes on the create user or modify user form, you can still submit a request without selecting a value. To make the user select a value for the required attribute, you must modify the request dataset to mark the attribute as mandatory. To do so:

1. When the administrative server and at least one Oracle Identity Manager managed server is running, login to Oracle Enterprise Manager Fusion Middleware Control by using the URL in the following format:

   http://*ADMINSTRATION_SERVER*:*PORT*/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:OIMAppMetadata**, **MDSAppRuntime**.

4. To export the request dataset:

   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. Specify the doc location as the following:

      `/metadata/iam-features-requestactions/model-data/CreateUserDataSet.xml`.

      `/metadata/iam-features-requestactions/model-data//ModifyUserDataset.xml`

      Note: Multiple documents can be set in the doc location while invoking operations exportMetaData or importMetaData.

   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. Edit the CreateUserDataSet.xml file, and change the value of the 'required' property to `true` for the attribute you created.

6. Edit the ModifyUserDataset.xml file, and change the value of the 'required' property to `true` for the attribute you created.

7. To import the request dataset:

   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. 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.

8. Restart Oracle Identity Manager.

7.10 Localizing Display Labels of UDFs

To localize display labels of UDFs:

1. Add a new custom field for the user object by referring to "Creating a Custom Attribute" on page 7-1 and ensure to publish the sandbox.
2. Export the BizEditorBundle.xlf file from MDS by referring to "Importing Metadata Files from MDS" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
3. Localize the content in BizEditorBundle.xlf to the expected locales. To do so:
   a. Create a copy of the BizEditorBundle.xlf file and rename it, for example, BizEditorBundle_zh_CN.xlf.
   b. Edit the <file> element from:

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

      To the following sample:

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

   c. Translate all the contents in the BizEditorBundle_zh_CN.xlf file.
4. Import the BizEditorBundle_zh_CN.xlf file to MDS by referring to "Exporting Metadata Files to MDS" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
5. Customize the Identity Self Service page to add the custom field label. See "Adding a Custom Attribute" on page 7-9 for details.
6. Switch the browser language to zh-CN, and log in to the Identity Self Service again.
7. Go to the page on which the custom attribute has been added, and confirm that the customized field label is using its localized value.

7.11 Configuring a Field as Mandatory Attribute in the Request Catalog

To configure a field as mandatory attribute in the request catalog:

1. In Oracle Identity Self Service, create and activate a sandbox. For detailed instructions on creating and activating a sandbox, see "Managing Sandboxes" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
2. On the left pane, under System Entities, click **Catalog**. The Catalog page is displayed.
3. Search for and select the application instance whose form page must be updated, and the click **Add to Cart**.
4. Click **Checkout**.
5. On the Cart Details page, under the Details section, the application instance form and its attributes are displayed.
6. Click **Customize**. The page opens in customization mode.

7. From the View menu, select **Source**. The object tree is displayed.
8. Under the Details section, select and click the attributes of the application instance form. A message confirming whether you want to edit the page is displayed.
9. Click **Edit**. In the object tree, the ADF component corresponding to the selection made in step 8 is selected.
10. Select the input text that is to be marked as mandatory, and click **Edit**. The Component Properties:inputText window opens.
11. Navigate to the required field, click the drop down icon adjacent to the field, select **Override**, and then select **Expression Builder**.
12. In the Expression Builder window, select the **Type a value or expression** option, and enter `true`.
13. Click **OK**, and then click **Apply**.
14. Click **OK** in the Component Properties:inputText. Click **Close** to quit customization mode.
15. Export the sandbox and publish it.

Part V

Application Management

This part describes application management in Oracle Identity Manager.

It contains the following chapters:

8

Managing IT Resources

IT resource is the target connectivity and connector configuration in an application instance. See "Managing Application Instances" on page 10-1 for information about application instances.

This chapter describes how to create and manage IT resources in the following sections:

8.1 Creating IT Resources

To create an IT resource:

> **Note:** The IT resource type is created before the IT resource can be created. The IT resource type can be created either by using the Design Console, or by importing the IT resource type using the Deployment Manager. See "IT Resources Type Definition Form" in *Developing and Customizing Applications for Oracle Identity Manager* for information about defining an IT resource type.

1. Login to Oracle Identity System Administration.
2. Under Provisioning Configuration, click **IT Resource**. The Manage IT Resource page is displayed.
3. Click **Create IT Resource**. The Create IT Resource wizard is displayed.
4. On the Step 1: Provide IT Resource Information page, enter the following information:
   - **IT Resource Name**: Enter a name for the IT resource.
   - **IT Resource Type**: Select an IT resource type for the IT resource.

     If you want to create an IT resource of the Remote Manager type, then select **Remote Manager** from the **IT Resource Type** list.
   - **Remote Manager**: If you want to associate the IT resource with a particular remote manager, then select the remote manager from this list. If you do not want to associate the IT resource with a remote manager, then leave this field blank.

Note: If you select **Remote Manager** from the **IT Resource Type** list, then you must not select a remote manager from the **Remote Manager** list.

5. Click **Continue**.
6. On the Step 2: Specify IT Resource Parameter Values page, specify values for the parameters of the IT resource, and then click **Continue**.
7. On the Step 3: Set Access Permission to IT Resource page, if you want to assign roles to the IT resource and set access permissions for the roles, then:

   a. Click **Assign Role**.

   b. For the roles that you want to assign to the IT resource, select **Assign** and the access permissions that you want to set. For example, if you want to assign the `ALL USERS` role and set the Read and Write permissions to this role, then you must select the respective check boxes in the row, as well as the Assign check box, for this role.

   c. Click **Assign**.
8. On the Step 3: Set Access Permission to IT Resource page, if you want to modify the access permissions of roles assigned to the IT resource, then:

   a. Click **Update Permissions**.

   b. Depending on whether you want to set or remove specific access permissions for roles displayed on this page, select or deselect the corresponding check boxes.

   Note: You cannot modify the access permissions of the `SYSTEM ADMINISTRATORS` role. You can modify the access permissions of only other roles that you assign to the IT resource.

   c. Click **Update**.
9. On the Step 3: Set Access Permission to IT Resource page, if you want to unassign a role from the IT resource, then:

   a. Select the **Unassign** check box for the role that you want to unassign.

   Note: You cannot unassign the `SYSTEM ADMINISTRATORS` role. You can unassign only other roles that you assign to the IT resource.

   b. Click **Unassign**.
10. Click **Continue**.
11. On the Step 4: Verify IT Resource Details page, review the information that you provided on the first, second, and third pages. If you want to make changes in the data entered on any page, click **Back** to revisit the page and then make the required changes.
12. To proceed with the creation of the IT resource, click **Continue**.
13. The Step 5: IT Resource Connection Result page displays the results of a connectivity test that is run using the IT resource information. If the test is

successful, then click **Create**. If the test fails, then you can perform one of the following steps:

- Click **Back** to revisit the previous pages and then make corrections in the IT resource creation information.
- Click **Cancel** to stop the procedure, and then begin from the first step onward.
- Proceed with the creation process by clicking **Continue**. You can fix the problem later, and then rerun the connectivity test.

Note: If no errors are encountered, then the label of the button is **Create**, not **Continue**.

---

14. Click **Finish**.

8.2 Managing IT Resources

To locate an IT resource:

1. In Oracle Identity System Administration, under Provisioning Configuration, click IT Resource. The Manage IT Resource page is displayed.
2. On the Manage IT Resource page, you can use one of the following search options to locate the IT resource that you want to view:
   - IT Resource Name: Enter the name of the IT resource, and then click **Search**.
   - IT Resource Type: Select the IT resource type of the IT resource, and then click **Search**.
   - Click **Search**.

On the Manage IT Resource page, the list of IT resources that meet the search criteria is displayed.

From this point onward, you can perform one of the following procedures on the IT resource:

- Viewing IT Resources
- Modifying IT Resources
- Deleting IT Resources

8.2.1 Viewing IT Resources

To view an IT resource:

1. From the list of IT resources displayed in the search results, click the IT resource name.
2. If you want to view the IT resource parameters and their values, then select **Details and Parameters** from the list at the top of the page. Similarly, if you want to view the administrative roles assigned to the IT resource, then select **Administrative Roles** from the list.

8.2.2 Modifying IT Resources

To modify an IT resource:

1. From the list of IT resources displayed in the search results, click the edit icon for the IT resource that you want to modify.
2. If you want to modify values of the IT resource parameters, then:
   a. Select **Details and Parameters** from the list at the top of the page.
   b. Make the required changes in the parameter values.
   c. To save the changes, click **Update**.
3. If you want to modify the administrative roles assigned to the IT resource, first select **Administrative Roles** from the list at the top of the page and then perform the required modification.
4. If you want to unassign an administrative role, select the **Unassign** check box in the row in which the role name is displayed and then click **Unassign**.

   Note:

   - When you click **Unassign**, the administrative roles that you select are immediately unassigned from the IT resource. You are not prompted to confirm that you want to unassign the selected administrative roles.
   - You cannot unassign the `SYSTEM ADMINISTRATORS` role.

5. If you want to assign new administrative roles to the IT resource, then:
   a. Click **Assign Role**.
   b. For the administrative roles that you want to assign to the IT resource, select the access permission check boxes and the **Assign** check box.
   c. Click **Assign**.
6. If you want to modify the access permissions of the administrative roles that are currently assigned to the IT resource, then:
   a. Click **Update Permissions**.
   b. Depending on the changes that you want to make, select or deselect the check boxes in the table.

      Note: You cannot change the access permissions of the `SYSTEM ADMINISTRATORS` role.

   c. To save the changes, click **Update**.

8.2.3 Deleting IT Resources

To delete an IT resource:

1. From the list of IT resources displayed in the search results, click the Delete icon for the IT resource that you want to delete.
2. To confirm that you want to delete the IT resource, click **Confirm Delete**.

   Note: Deleting IT resource instances soft-deletes the corresponding application instances.

9

Managing Generic Connectors

Generic connectors are managed by using Oracle Identity System Administration. See "Predefined Providers for Generic Technology Connectors" in *Developing and Customizing Applications for Oracle Identity Manager* for information about generic technology connectors (GTC).

This chapter describes how to create and manage generic connectors in the following sections:

- Creating Generic Technology Connectors
- Managing Generic Technology Connectors

9.1 Creating Generic Technology Connectors

This section explains how to create generic technology connectors.

The procedure to create a generic technology connector is composed of the following steps:

- Determining Provider Requirements
- Selecting the Providers to Include
- Addressing the Prerequisites
- Using Identity System Administration to Create the Connector
- Configuring Reconciliation
- Configuring Provisioning
- Creating the Form and Publishing the Application Instance
- Enabling Logging

9.1.1 Determining Provider Requirements

The following providers can be used as the building blocks of the generic technology connectors you create:

- Reconciliation Transport Provider
- Reconciliation Format Provider
- Provisioning Transport Provider
- Provisioning Format Provider

- Transformation Provider
- Validation Provider

Based on your knowledge of the data formats and data transport mechanisms supported by the target system, identify the providers that must be included in the generic technology connector that you create. If the target system supports multiple data formats and data transport mechanisms, you must select a single combination of the transport and format providers discussed in the first chapter. You cannot include, for example, multiple reconciliation format providers in a single generic technology connector.

9.1.2 Selecting the Providers to Include

Identify the predefined providers that can be used to meet your provider requirements. See "Predefined Providers for Generic Technology Connectors" in *Developing and Customizing Applications for Oracle Identity Manager* for information about the predefined providers.

If all your provider requirements are addressed by the predefined providers, you need not create custom providers. You must create custom providers to address only the requirements that are not addressed by the predefined providers.

9.1.3 Addressing the Prerequisites

You must address the following prerequisites:

- If you are creating the generic technology connector on a production server, enable the cache for the following cache categories:
  - `GenericConnector`
  - `GenericConnectorProviders`
- Testing connectivity between the target system server and the Oracle Identity Manager server

  You must take steps to ensure that connectivity can be established between the target system server and the Oracle Identity Manager server. For example, in a UNIX environment, you must enter the fully qualified host name of the Oracle Identity Manager server in the `/etc/hosts` file on the target system server.
- Creating the user account to be used for creating the generic technology connector

  All users belonging to the `SYSTEM ADMINISTRATORS` group of Oracle Identity Manager can create generic technology connectors. Alternatively, members of a group to which you assign the required menu items and permissions can create generic technology connectors.

  The required menu items are as follows:
  - Create Generic Technology Connector menu item
  - Manage Generic Technology Connector menu item

  The required permissions are as follows:
  - Form Designer (Allow Insert, Write Access, Delete Access)
  - Structure Utility. Additional Column (Allow Insert, Write Access, Delete Access)
  - Meta-Table Hierarchy (Allow Insert, Write Access, Delete Access)

If these permissions are not correctly assigned to the group, an error is thrown when the user clicks the Create button on the final Identity System Administration page for creating generic technology connectors.

Note: In an Oracle Identity Manager deployment that is integrated with Access Manager (OAM), the OIMSignatureAuthenticator authentication provider is not configured by default. If you use Oracle Identity Manager 9.x connectors, such as GTC, or if your custom code uses signature-based OIMClient login, then you must enable the OIMSignatureAuthenticator authentication provider.

For information about enabling OIMSignatureAuthenticator, see "OIMSignatureAuthenticator Not Configured for Oracle Identity Manager Domain Security Realm" in the *Oracle Fusion Middleware Release Notes*.

9.1.4 Using Identity System Administration to Create the Connector

To navigate to the first Identity System Administration page for creating a generic technology connector, login to Identity System Administration, and click **Generic Connector** under Provisioning Configuration. In the Manage Connectors page, click **Create**.

From this point onward, page-wise instructions are provided in the following sections:

- Step 1: Provide Basic Information Page
- Step 2: Specify Parameter Values Page
- Step 3: Modify Connector Configuration Page
- Step 4: Verify Connector Form Names Page
- Step 5: Verify Connector Information Page

9.1.4.1 Step 1: Provide Basic Information Page

To provide basic information about the generic technology connector that you want to create, use this page as follows

1. In the **Name** field, specify a name for the generic technology connector.

   The following are guidelines related to selecting a name for the generic technology connector:

   - The name must not be the same as that of any other connector (predefined connector or generic technology connector) on this Oracle Identity Manager installation.
   - The name must not be the same as that of any other connector object (such as resource objects, IT resources, and process forms) on this Oracle Identity Manager installation.

   Note: An error message is displayed if you specify a name that is the same as the name of an existing connector. However, an error message is *not* displayed if you specify a name that is the same as the name of an existing connector object. Therefore, you must ensure that the name you want to specify is not the same as the name of any existing connector object.

- The name must not contain non-ASCII characters, because Oracle Identity Manager does not support non-ASCII characters in connector names. However, you can include the underscore character (_) in the name.

2. If you want to use the generic technology connector for reconciliation, select **Reconciliation** and perform the following steps:

   - From the **Transport Provider** list, select the reconciliation transport provider that you want to use for this connector. This list displays the predefined reconciliation transport providers and the reconciliation transport providers that you create.
   - From the **Format Provider** list, select the reconciliation format provider that you want to use for this connector. This list displays the predefined reconciliation format providers and the reconciliation format providers that you create.

     Note: If you select the shared drive reconciliation transport provider, you must also select the CSV reconciliation format provider because all the parameters of this provider are bundled with the parameters of the shared drive reconciliation transport provider.

   - If you want to use the connector to perform trusted source reconciliation with the target system, select **Trusted Source Reconciliation**.

     Note: If you select the Trusted Source Reconciliation check box, the Provisioning region of the page is disabled. This is because you cannot provision to a target system that you designate as a trusted source. You can only reconcile data from a trusted source.

3. If you want to use the generic technology connector for provisioning, select **Provisioning** and perform the following steps:

   Note: You can select only Reconciliation, only Provisioning, or both Reconciliation and Provisioning.

   - From the **Transport Provider** list, select the provisioning transport provider that you want to use for this connector. This list displays the predefined provisioning transport providers and the provisioning transport providers that you create.

     If you select the Web Services provisioning transport provider and if Secure Sockets Layer (SSL) is enabled for the target Web service, you must perform the procedure described in "Configuring SSL Communication Between Oracle Identity Manager and the Target System Web Service" in *Developing and Customizing Applications for Oracle Identity Manager*.

   - From the **Format Provider** list, select the provisioning format provider that you want to use for this connector. This list displays the predefined provisioning format providers and the provisioning format providers that you create.

     If you select the SPML provisioning format provider, you must also select the Web Services provisioning transport provider because the parameters of this

provider are related to the parameters of the Web Services provisioning transport provider.

4. Click **Continue**.

Table 9–1 lists sample entries for the GUI elements on the Step 1: Provide Basic Information page.

Table 9–1 Sample Entries for the Step 1: Provide Basic Information Page

Label on the Step 1: Provide Basic Information Page	Sample Value or Action	Reference Information
Name field	MyGTC2	NA
Reconciliation check box	Check box selected	NA
Transport Provider list	Shared Drive	shared drive reconciliation transport provider
Format Provider list	CSV	CSV Reconciliation format provider
Provisioning check box	Check box selected	NA
Transport Provider list	Web Services	Web Services provisioning transport provider
Format Provider list	SPML	SPML provisioning format provider

9.1.4.2 Step 2: Specify Parameter Values Page

Use this page to specify values for the parameters of the providers that you select on the Step 1: Provide Basic Information page.

On this page, the provider parameters are divided into two categories:

- Run-time parameters

  See Also: "Predefined Providers for Generic Technology Connectors" in *Developing and Customizing Applications for Oracle Identity Manager* for detailed information about the run-time parameters of predefined providers that you select on the Step 1: Provide Basic Information page

  Run-time parameters are input variables of the providers that you select on the previous page. A run-time parameter represents a value that is not constrained by the design of the provider. For example, the location of the directories containing the data files that you want to reconcile is a run-time parameter.

- Design parameters

  The parameters listed in this section are either design parameters of providers or reconciliation-specific parameters that are common to all generic technology connectors. A design parameter represents a value or set of values that is defined as part of the provider design.

  See Also: "Predefined Providers for Generic Technology Connectors" in *Developing and Customizing Applications for Oracle Identity Manager* for detailed information about the design parameters of predefined providers that you select on the Step 1: Provide Basic Information page

  For example:

The format of data files that can be parsed by a format provider is a design parameter for that provider. While designing the provider, you define the set of formats the provider can parse. On the Step 2: Specify Parameter Values page, you specify the particular format (from the set of supported formats) that an instance of the format provider must parse.

The following are reconciliation-specific design parameters:

Note: If you do not select the Reconciliation option on the previous page, these reconciliation-specific design parameters are not displayed on this page.

– **Batch Size**

  Use this parameter to specify a batch size for the reconciliation run. By using this parameter, you can break into batches the total number of records that the reconciliation engine fetches from the target system during each reconciliation run.

  The default value of this parameter is `All`.

– **Stop Reconciliation Threshold**

  During reconciliation, data from the reconciliation format provider is accepted as input by the validation provider. Some of the reconciliation data records may not clear the validation checks. You can use the Stop Reconciliation Threshold parameter to automatically stop reconciliation if the percentage of records that fail the validation checks to the total number of reconciliation records processed exceeds the specified value.

  The following example illustrates how this parameter works:

  Suppose you specify 20 as the value of the Stop Reconciliation Threshold parameter. This means that you want reconciliation to stop if the percentage of failed records to the total number of records processed becomes equal to or greater than 20. Suppose the second and eighth records fail the validation checks. At this stage, the number of failed records is 2 and the total number of records processed is 8. The percentage of failed records is 25, which is greater than the specified threshold of 20. Therefore, reconciliation is stopped after the eighth record is processed.

  Note:

  - The Stop Reconciliation Threshold parameter is used during reconciliation only if you select validation Providers on the Step 3: Modify Connector Configuration page.
  - If reconciliation is stopped because the actual percentage of failed records exceeds the specified percentage, the records that have already been reconciled into Oracle Identity Manager are not removed.

  The default value of this parameter is `None`. This default value specifies that during a reconciliation run, you want all the target system records to be processed, regardless of the number of records that fail the checks.

– **Stop Threshold Minimum Records**

If you use the Stop Reconciliation Threshold parameter, there may be a problem if invalid records are encountered right at the beginning of the reconciliation run. For example, suppose you specify 40 as the value of the Stop Reconciliation Threshold parameter. When reconciliation starts, suppose the first record fails the validation checks. At this stage, the percentage of failed records to total records processed is 100. Therefore, reconciliation would stop immediately after the first record is processed.

To avoid such situations, you can use the Stop Threshold Minimum Records parameter in conjunction with the Stop Reconciliation Threshold parameter. The Stop Threshold Minimum Records parameter specifies the number of records that must be processed by the validation provider before the Stop Reconciliation Threshold validation is enabled.

The following example illustrates how this parameter works:

Suppose you specify the following values:

Stop Reconciliation Threshold: 20

Stop Threshold Minimum Records: 80

With these values, from the eighty-first record onward, the Stop Reconciliation Threshold validation is enabled. In other words, after the eightieth record is processed, if any record fails the validation check, the reconciliation engine calculates the percentage of failed records to total records processed.

The default value of this parameter is `None`.

Note:

- The Stop Threshold Minimum Records parameter is used during reconciliation only if you select validation Providers on the Step 3: Modify Connector Configuration page.
- You must specify a value for the Stop Threshold Minimum Records parameter if you specify a value for the Stop Reconciliation Threshold parameter.

– **Reconciliation Type**

Use this parameter to specify whether you want the reconciliation engine to perform incremental or full reconciliation.

Note: The outcome of both full and incremental reconciliation is the same: target system records that are created or updated after the last reconciliation run are reconciled into Oracle Identity Manager.

In incremental reconciliation, only target system records that are newly added or modified after the last reconciliation run are brought to Oracle Identity Manager. Reconciliation events are created for each of these records.

In full reconciliation, all target system records are brought to Oracle Identity Manager. The optimized reconciliation feature identifies and ignores records that have already been reconciled. Reconciliation events are created for the remaining records.

You must select incremental reconciliation if either one of the following conditions is true:

* The target system time stamps or uniquely marks (in some way) files or individual data records that it generates, and the reconciliation transport provider can recognize records that have been time stamped or marked by the target system.

  For example:

  Suppose the target system can time stamp the creation of or modifications to user data records. If you can create a custom reconciliation transport provider that can read this time-stamp information, only new or modified data records is transported to Oracle Identity Manager during reconciliation.

* The target system provides only data records that are newly added or modified after the last reconciliation run.

If *neither* of these conditions is true, you must select full reconciliation.

– **Reconcile Deletion of Multivalued Attribute Data**

Use this parameter to specify whether or not you want to reconcile into Oracle Identity Manager the deletion of multivalued attribute data (child data) on the target system.

The following example explains how this design parameter works:

There is an account for user `John Doe` on the target system. This user is a member of two user groups, `CREATE USERS` and `REVIEW PERMISSIONS`, on the target system. This user account (along with the group membership information) also exists on Oracle Identity Manager.

On the target system, suppose this user is removed from the `REVIEW PERMISSIONS` group. During the next reconciliation run, the action that is taken in Oracle Identity Manager depends on whether or not you select the **Reconcile Deletion of Multivalued Attribute Data** check box:

* If you select the check box, information about this user being a member of the `REVIEW PERMISSIONS` group on the target system is removed from the Oracle Identity Manager database. All other changes made to this user account on the target system are also reconciled.

* If you do not select the check box, information about this user being a member of the `REVIEW PERMISSIONS` group on the target system is *not* removed from the Oracle Identity Manager database. However, all other changes made to this user account on the target system are reconciled.

– **Source Date Format**

Use this parameter to specify the format in which date values are stored in the target system.

The format that you specify is used to validate date values fetched during reconciliation and to convert the date values to the format used internally by Oracle Identity Manager.

The Validate Date Format provider is one of the predefined validation providers. During a reconciliation run, the Validate Date Format provider uses the source date format to validate date values fetched from the target system. Only date values that match the source date format are converted to the date format used by Oracle Identity Manager and reconciled. This format validation and conversion applies to all date fields (for example, Date of Birth and Hire Date) of the target system.

For information about the date formats that you can specify, see the following page on the Sun Java Web site:

http://java.sun.com/docs/books/tutorial/i18n/format/simpleDateFormat.html

Note: If you want the source date format to be used in date validation, while performing the procedure described in "Adding or Editing Fields in Data Sets" on page 9-18, you must:

- Map date fields of the Source data sets to date fields of the reconciliation staging data sets.
- Edit each date field of the reconciliation staging data sets and set its data type to the Date data type.

The default value of the Source Date Format parameter is the date format specified as the value of the `XL.DefaultDateFormat` system property. If you do not specify a value for the Source Date Format parameter, the default date format is used for date validation during reconciliation.

See Also: "System Properties in Oracle Identity Manager" in the *Oracle Fusion Middleware Administering Oracle Identity Manager* for information about the system properties of Oracle Identity Manager

The following example illustrates how the Source Date Format parameter is used:

Suppose the following are date values in the target system:

– Date 1: `05/04/2007 06:25:44 PM`

– Date 2: `05/06/2007 07:31:44 PM`

– Date 3: `Thu, Apr 9, '98`

– Date 4: `07/03/2008 02:15:55 PM`

Scenario 1:

While creating the connector, you had entered the following as the value of the Source Date Format parameter:

`MM/dd/yyyy hh:mm:ss a`

During a reconciliation run, the record containing the Date 3 value is not reconciled because it does not conform to the specified source date format.

Scenario 2:

While creating the connector, you had not entered a value for the Source Date Format parameter. Therefore, during a reconciliation run, all four records are validated against the date format specified as the value of the `XL.DefaultDateFormat` system property.

The following is a provisioning-specific design parameter:

Note: If you do not select the Provisioning option on the previous page, this provisioning-specific design parameter is not displayed.

- **Target Date Format**

  Use this parameter to specify the format in which you want to send date values to the target system during provisioning operations.

  During a provisioning operation, date values are converted to the format that you specify as the value of the Target Date Format parameter. This format conversion applies to all date fields (for example, Date of Birth and Hire Date) that are used in the provisioning operation.

  For information about the date formats that you can specify, see the following page on the Sun Java Web site:

  `http://java.sun.com/docs/books/tutorial/i18n/format/simpleDateFormat.html`

  If you do not specify a date format, the following date format is used as the default value of this parameter:

  `yyyy/MM/dd hh:mm:ss z`

  The following example illustrates how the Target Date Format parameter is used:

  During a provisioning operation, any date value that you enter is in the `yyyy/MM/dd hh:mm:ss z` format.

  Scenario 1:

  While creating the connector, you had entered the following as the value of the Target Date Format parameter:

  `yyyy.MM.dd G 'at' hh:mm:ss z`

  During a provisioning operation, an Oracle Identity Manager date value (for example, 2007/05/04 06:25:44 IST) is converted into the target date format (for example, 2007.05.04 AD at 06:25:44 IST) and sent to the target system.

  Scenario 2:

  While creating the connector, you had not entered a value for the Target Date Format parameter. During a provisioning operation, date values are sent to the target system in the (default) `yyyy/MM/dd hh:mm:ss z` format.

After you specify values for the run-time and design parameters, click **Continue**.

Note: If any value that you provide on this page is not correct, an error message is displayed at the top of the page after you click **Continue**. If this happens, fix the parameter value and click **Continue** again.

Table 9–2 lists sample entries for the Step 2: Specify Parameter Values page. The GUI elements displayed on this page are based on the entries made on the Step 1: Provide Basic Information page.

Table 9–2 Sample Entries for the Step 2: Specify Parameter Values Page

Label on the Step 2: Specify Parameter Values Page	Sample Value or Action	Reference Information
Run-Time Parameters of the Shared Drive Reconciliation Transport Provider		"Shared Drive Reconciliation Transport Provider" in *Developing and Customizing Applications for Oracle Identity Manager*
Staging Directory (Parent Identity Data) field	D:\gctestdata\commaDelimited\parent	NA
Staging Directory (Multivalued Identity Data) field	D:\gctestdata\commaDelimited\child	NA
Archiving Directory field	D:\gctestdata\commaDelimited\archive	NA
File Prefix field	file	NA
Specified Delimiter field	,	NA
Tab Delimiter check box	Check box not selected	NA
Fixed Column Width field		NA
Unique Attribute (Parent Data) field	UserIDTD	NA
Run-Time Parameter of the Web Services Provisioning Transport Provider		"Web Services Provisioning Transport Provider" in Developing and Customizing Applications for Oracle Identity Manager
Web Service URL field	http://acme123:8080/spmlws/services/HttpSoap11	NA
Run-Time Parameters of the SPML Provisioning Format Provider		
Target ID field	target	NA
User Name (authentication) field	xelsysadm	NA
User Password (authentication) field		NA
Design Parameters of the Shared Drive Reconciliation Transport Provider		"Shared Drive Reconciliation Transport Provider" in *Developing and Customizing Applications for Oracle Identity Manager*
File Encoding field	Cp1251	NA
Design Parameters of the Web Services Provisioning Transport Provider		"Web Services Provisioning Transport Provider" in Developing and Customizing Applications for Oracle Identity Manager
Web Service SOAP Action field	http://xmlns.oracle.com/OIM/provisioning/processRequest	NA
Design Parameters of the SPML Provisioning Format Provider		
WSSE Configured for SPML Web Service? check box	Check box not selected	NA

Table 9–2 (Cont.) Sample Entries for the Step 2: Specify Parameter Values Page

Label on the Step 2: Specify Parameter Values Page	Sample Value or Action	Reference Information
Custom Authentication Credentials Namespace field	http://xmlns.oracle.com/OIM/provisioning	NA
Custom Authentication Header Element field	OIMUser	NA
Custom Element to Store User Name field	OIMUserId	NA
Custom Element to Store Password field	OIMUserPassword	NA
SPML Web Service Binding Style (DOCUMENT or RPC) field	RPC	NA
SPML Web Service Complex Data Type field		NA
SPML Web Service Operation Name field	processRequest	NA
SPML Web Service Target Namespace field	http://xmlns.oracle.com/OIM/provisioning	NA
SPML Web Service Soap Message Body Prefix field		NA
ID Attribute for Child Dataset Holding Group Membership Information field		NA
Generic Design Parameters		NA
Target Date Format field	yyyy-MM-dd hh:mm:ss.fffffffff	NA
Batch Size field	All	NA
Stop Reconciliation Threshold field	None	NA
Stop Threshold Minimum Records field	None	NA
Source Date Format field	yyyy/MM/dd hh:mm:ss z	NA
Reconcile Deletion of Multivalued Attribute Data check box	Check box selected	NA
Reconciliation Type list	Incremental	NA

9.1.4.3 Step 3: Modify Connector Configuration Page

Use this page to define data sets and mappings between the fields of the data sets. In other words, you use this page to specify the user data fields that you want to:

- Propagate from the target system to Oracle Identity Manager during reconciliation
- Propagate from Oracle Identity Manager to the target system during provisioning

In the generic technology connector context, the term **metadata** refers to the set of identity fields that constitute the user account information on the target system.

First Name, Last Name, Hire Date, and Department ID are examples of user data fields that constitute metadata. The values assigned to these fields constitute the user data on the target system. For example, the identity information of user John Doe on the target system can be composed of the following fields:

- First Name: John
- Last Name: Doe
- Hire Date: 04-December-2007

- Department ID: Sales
- . . .

After you click the **Continue** button on the Step 2: Specify Parameter Values page, the metadata displayed on the Step 3: Modify Connector Configuration page depends on the following factors:

- Input provided on the Step 1: Provide Basic Information and Step 2: Specify Parameter Values pages
- Availability of sample target system data

> **Note:** In the generic technology connector context, the term **metadata detection** refers to the process in which sample user data is read from the target system and the corresponding metadata (identity field names) is displayed on the Step 3: Modify Connector Configuration page.

Oracle Identity Manager performs the following steps while attempting to detect metadata:

1. The reconciliation transport provider and reconciliation format provider try to fetch and parse metadata from the target system.

   Together, the shared drive reconciliation transport provider and CSV reconciliation format provider can detect metadata from the target system. If you want custom providers to perform the same function, you must ensure that:

   - The Java code for the reconciliation transport provider contains an implementation of the `getMetadata()` method of the `ReconTransportProvider` interface.
   - The Java code for the reconciliation format provider contains an implementation of the `parseMetadata()` method of the `ReconFormatProvider` interface.

   If these providers successfully fetch and parse metadata from the target system, Oracle Identity Manager uses information returned by them to display metadata and the following step is not performed.

2. If the reconciliation transport provider and reconciliation format provider cannot fetch and parse metadata from the target system, the provisioning transport provider and provisioning format provider try to perform this function.

   The Web Services provisioning transport provider and SPML provisioning format provider cannot detect metadata from the target system. If you want custom providers to be able to detect metadata, you must ensure that:

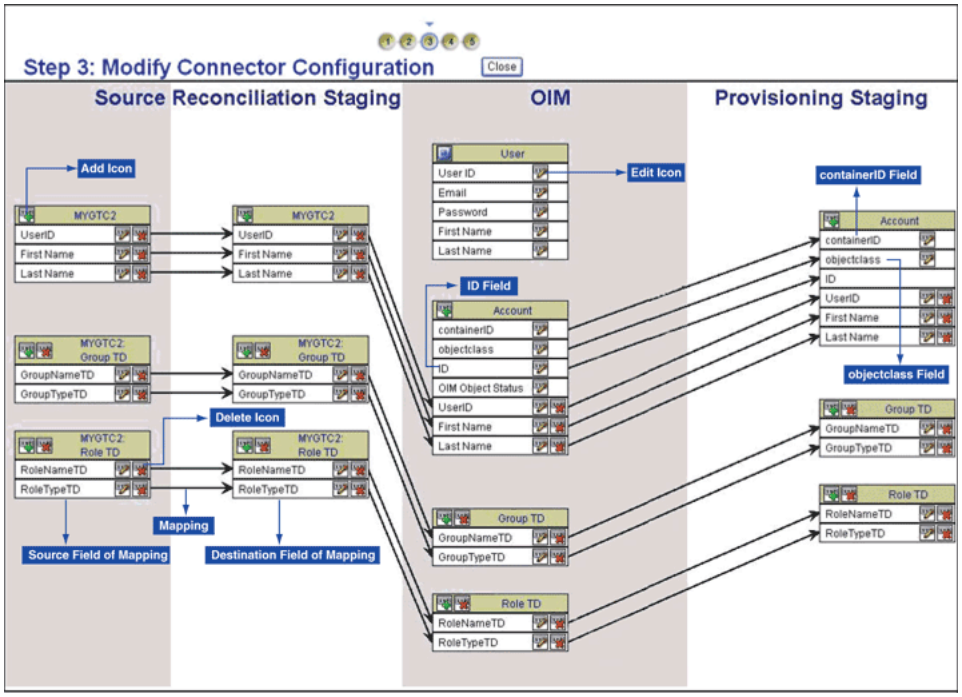
   - The Java code for the provisioning transport provider contains an implementation of the `defineMetadata()` method of the `ProvisioningTransportProvider` interface.
   - The Java code for the provisioning format provider contains an implementation of the `parseMetadata()` method of the `ProvisioningFormatProvider` interface.

   If the provisioning transport provider and provisioning format provider successfully fetch and parse metadata from the target system, Oracle Identity Manager uses information returned by these providers to display metadata. If these providers are not successful, only the default fields defined for any of the

provisioning-specific providers that you select are displayed. For example, the `ID` field of the OIM - Account data set and the `objectClass` and `containerID` fields of the provisioning staging data set are displayed by default. These data sets and fields are discussed later in this guide.

Figure 9–1 shows the Step 3: Modify Connector Configuration page for the sample entries listed at the end of the "Step 1: Provide Basic Information Page" and "Step 2: Specify Parameter Values Page" sections.

Figure 9–1 Step 3: Modify Connector Configuration Page



- Data Sets
- Mappings

Data Sets

The data sets displayed on the Step 3: Modify Connector Configuration page are categorized as follows:

- Source

  The Source data sets are displayed only if you select the Reconciliation option on the first page, regardless of whether or not you select the Provisioning option.

- Reconciliation Staging

  The reconciliation staging data sets are displayed only if you select the Reconciliation option on the Step 1: Provide Basic Information page, regardless of whether or not you select the Provisioning option.

- Oracle Identity Manager

  The Oracle Identity Manager data sets are always displayed, regardless of the options you select on the Step 1: Provide Basic Information page. However, the OIM - Account data set and its child data sets are not displayed if you select the

Trusted Source Reconciliation option on the Step 1: Provide Basic Information page. To overcome this issue, you must perform the following steps:

1. Open the generic technology connector and navigate to Jgraph screen.
2. In the Reconciliation Staging of the Jgraph screen, modify the field data type to **Date** for all the fields which holds date value.
3. Save the connector.

The fields displayed in the OIM - User data set are predefined for the Oracle Identity Manager User. You can show or minimize the full list of OIM - User data set fields by clicking the arrow icon at the top of the data set. The following fields are displayed in the minimized state of the data set:

– User ID

– Email

– Password

– First Name

– Last Name

Note: If you select the Trusted Source Reconciliation option on the Step 1: Provide Basic Information Page, all the fields of the OIM - User data set are displayed and you cannot use the arrow icon to minimize the display.

These fields constitute the minimum set of Oracle Identity Manager User fields for which values must be defined. You can designate some or all of the remaining OIM - User data set fields as mandatory Oracle Identity Manager User fields for your Oracle Identity Manager installation. You do this by ensuring that these fields always hold values when the Oracle Identity Manager User is created.

Note: Data set and field names that take up more than a certain amount of space are truncated and dots are displayed after the truncated part of the names. For example, the Deprovisioning Date field of the OIM - User data set is displayed as follows:

```
Deprovisioning Da..
```

To view the full name of a field, you can click the edit icon for that field or the field to which that field is mapped. In the pop-up window, the field name that you want to view is on either the first page or the second page, depending on the data set to which the field belongs.

You can add user-defined fields (UDFs) to the list of predefined Oracle Identity Manager User fields by using the Design Console. These UDFs are displayed in the OIM - User data set on the Step 3: Modify Connector Configuration page.

Depending on the options that you select on the Step 1: Provide Basic Information page, some fields are displayed by default on the Step 3: Modify Connector Configuration page:

– `ID` field

  The `ID` field is displayed by default in the OIM - Account data set, regardless of whether or not you select the Reconciliation option or Provisioning option

on the Step 1: Provide Basic Information page. When an account is created, this field is used to store the value that uniquely identifies the account in Oracle Identity Manager and in the target system. For a particular user, this unique field is used to direct other operations, such as modify, delete, enable, disable, and child data operations.

Every target system would have a unique field for tracking the creation of and updates made to a user account. While creating a custom provisioning transport provider, you must ensure that the provider retrieves this unique field value from the target system at the end of a Create User operation. This value must be used to populate the `ID` field of the OIM - Account data set.

During reconciliation, the value of the `ID` field must come from the corresponding unique field of the reconciliation staging data set. To set this up, you must create a mapping between the two fields. The procedure to create a mapping is discussed later in this section.

Caution: If you select both the Provisioning and Reconciliation options while creating a generic technology connector and if you do not create a mapping between the ID field and the unique field of the target system, records that are linked through reconciliation cannot be used for provisioning operations (such as modify, delete, enable, disable, and child data operations). This is because the ID field is not populated in the linked records.

– `objectClass` field

The `objectClass` field is displayed by default in the OIM - Account data set and provisioning staging data set only if you select the SPML provisioning format provider on the Step 1: Provide Basic Information page.

– `containerID` field

The `containerID` field is displayed by default in the OIM - Account data set and provisioning staging data set only if you select the SPML provisioning format provider on the Step 1: Provide Basic Information page.

- Provisioning Staging

The provisioning staging data sets are displayed only if you select the Provisioning option on the first page, regardless of whether or not you select the Reconciliation option.

The display of data sets on the Step 3: Modify Connector Configuration page depends on the input that you provide on the Step 1: Provide Basic Information page and Step 2: Specify Parameter Values page. The display of fields within the data sets depends on whether or not metadata detection has taken place.

Note: Metadata detection does not take place if any of the following conditions are true:

- Sample target system data (including metadata) is not available.
- The Transport and format providers that you select are not capable of detecting metadata from sample target system data.

This is illustrated by the following example:

Suppose you select only the Reconciliation option on the Step 1: Provide Basic Information page. In addition, metadata detection has not taken place. Under these conditions, the display of data sets and fields on the Step 3: Modify Connector Configuration page can be summarized as follows:

The following data sets are displayed:

- Source
- Reconciliation Staging
- Oracle Identity Manager

The fields that constitute the data sets are *not* displayed.

In addition, if you had selected the Trusted Source Reconciliation option on the Step 1: Provide Basic Information page, the OIM - Account data set and its child data sets are not displayed.

In Table 9–3, Scenario 1 shows the outcome of this set of input conditions. The rest of the scenarios in this table describe the display of data sets and fields under the combination of input conditions listed in the first row and first column of the table.

Table 9–3 Display of Data Sets and Fields Under Various Input Conditions

Metadata Detection	Only Reconciliation Option Selected	Both Reconciliation and Provisioning Options Selected	Only Provisioning Option Selected
Metadata detection has *not* taken place	**Scenario 1** The following data sets are displayed: ■ Source ■ Reconciliation Staging ■ Oracle Identity Manager The fields that constitute the data sets are *not* displayed. If you select the Trusted Source Reconciliation option on the Step 1: Provide Basic Information page, the OIM - Account data set and its child data sets are not displayed.	**Scenario 2** The following data sets are displayed: ■ Source ■ Reconciliation Staging ■ Oracle Identity Manager ■ Provisioning Staging The fields that constitute the data sets are *not* displayed.	**Scenario 3** The following data sets are displayed: ■ Oracle Identity Manager ■ Provisioning Staging The fields that constitute the data sets are *not* displayed.
Metadata detection has taken place	**Scenario 4** The following data sets are displayed: ■ Source ■ Reconciliation Staging ■ Oracle Identity Manager The fields that constitute the data sets are displayed. If you select the Trusted Source Reconciliation option on the Step 1: Provide Basic Information page, the OIM - Account data set and its child data sets are not displayed.	**Scenario 5** The following data sets are displayed: ■ Source ■ Reconciliation Staging ■ Oracle Identity Manager ■ Provisioning Staging The fields that constitute the data sets are displayed.	**Scenario 6** The following data sets are displayed: ■ Oracle Identity Manager ■ Provisioning Staging The fields that constitute the data sets are displayed.

Mappings

Each flow line displayed on the Step 3: Modify Connector Configuration page represents a mapping (link) between two fields of different data sets. A mapping serves one of the following purposes:

- Establishes a data flow path between fields of two data sets, for either provisioning or reconciliation

  A mapping of this type forms the basis for validations or transformations to be performed on data.

- Creates a basis for comparing (matching) field values of two data sets

  The following are examples of matching-only mappings:

  – Mappings created between fields of the reconciliation staging data set and the OIM - User data set form the basis of a reconciliation rule.

  – A mapping between the unique field of the reconciliation staging data set and the `ID` field of the OIM - Account data set helps identify the key field for reconciliation matching. Along with the ID field, other fields of the OIM - Account data set can be (matching-only) mapped to corresponding fields of the reconciliation staging data set to create a composite key field for reconciliation matching.

You can perform the following actions on the Step 3: Modify Connector Configuration page:

- Adding or Editing Fields in Data Sets
- Removing Fields from Data Sets
- Removing Mappings Between Fields
- Removing Child Data Sets

9.1.4.3.1 Adding or Editing Fields in Data Sets Identity fields detected through metadata detection are displayed on the Step 3: Modify Connector Configuration page. You can modify these fields and the mappings between them. If required, you can also add new fields on this page and create mappings between them.

The following is a summary of the actions that you can perform while adding or editing fields on the Step 3: Modify Connector Configuration page:

> **Note:** These actions are described in detail in the procedure that follows this list. The procedure also describes the conditions that must be fulfilled before you can perform some of these actions.

- Default attributes (such as the data type and length) are assigned to the fields displayed through metadata detection. You must edit these fields to set the required attributes for them.

> **Note:** Oracle Identity Manager can recognize date values fetched during reconciliation only if you set the Date data type for fields of the reconciliation staging data sets. In addition, if you have specified a value for the Source Date Format parameter on the Step 2: Specify Parameter Values page, you must map date fields of the Source data sets to the corresponding date fields of the reconciliation staging data sets.

- You can create transformation mappings between fields by using a transformation provider. While performing this action, you can use the predefined concatenation transformation provider or translation transformation provider, or a custom transformation provider that you have created.
- You can create matching-only mappings between fields of the reconciliation staging data set and Oracle Identity Manager data sets. Matching-only mappings that you create between the reconciliation staging data set and the OIM - User data set forms the reconciliation rule. Matching-only mappings that you create between the reconciliation staging data set and the OIM - Account data set identifies the key field for reconciliation matching.
- You can add a child data set to an existing data set.
- You can encrypt the value of a field, both in the process form and in the database.
- You can designate a field as a lookup field and select an input source for the field. The input source can be a lookup definition or a combination of columns from Oracle Identity Manager database tables.
- You can configure user account status reconciliation.

  If you want to configure user account status reconciliation, refer to the "Configuring Account Status Reconciliation" section.

To add or edit a field in a data set:

> **Note:** The display of the GUI elements and pages described in the following steps depends on the data set in which you are adding or editing a field. For example, the Required and Encrypted check boxes are not displayed if you are adding or editing a field in a Source data set.

1. Depending on whether you want to add or edit a field, click the Add icon for the data set or the edit icon for the field.
2. On the Step 1: Field Information page, specify values for the following GUI elements:
   - **Field Name**: If you are adding a field, specify a name for the field. The field name that you specify must contain only ASCII characters, because non-ASCII characters are not allowed.
   - **Mapping Action**: Select the type of mapping that you want to create with this field as the destination field of the mapping. You can select one of the following mapping actions:
     - Select **Create Mapping Without Transformation** if you only want to create a one-to-one mapping between a source (input) field and the field that you are adding or editing, and you do not want to use a transformation provider.
     - Select the **Remove Mapping** option if you are editing the field and you want to remove the mapping for which this field is the destination field. The procedure to remove a mapping is covered in detail in the Removing Mapping Between Fields section.
     - The transformation mapping options displayed in the Mapping Action list are based on the predefined transformation providers and the custom

transformation providers that you create. The following menu options correspond to the predefined transformation providers:

*** Create Mapping With Concatenation**

*** Create Mapping With Translation**

Apply the following guidelines while selecting a transformation mapping:

* You can create transformation mappings only between fields of the following data sets:

- Source and Reconciliation Staging

- Oracle Identity Manager and Provisioning Staging

This means that, for example, you cannot create transformation mappings between a field in a reconciliation staging data set and a field in an Oracle Identity Manager data set.

You cannot create a 1-to-2 mapping with the following source and destination fields:

Source field: Unique field of the reconciliation staging data

Destination fields: `User ID` field of the OIM - User data set and `ID` field of the OIM - Account data set

This mapping is not supported. Instead, you must create a one-to-one mapping between the unique field of the reconciliation staging data and either the `User ID` field (of the OIM - User data set) or the `ID` field (of the OIM - Account data set).

* Ensure that all the fields of provisioning staging data sets are mapped to corresponding fields of OIM - User and OIM - Account data sets.

* When you create a mapping that has any field of the OIM - User data set as the source or destination field, the display of the OIM - User data set fields list is frozen in the position it was in (expanded or minimized) when the mapping was created. To unfreeze the display of the OIM - User data set so that you are able to use the arrow icon, you must remove all mappings that have any OIM - User data set field as the source or destination field.

* A literal field can be used as one of the input fields of a transformation field. If you select the Literal option, you must enter a value in the field. You must not leave the field blank after selecting it.

- **Matching Only**: Select this check box if the field is to be used as the destination field of a matching-only mapping. As mentioned earlier in this document, you can create the following types of matching-only mappings:

Note: You must create matching-only mappings for both parent and child data sets.

  – To create the reconciliation rule, you create matching-only mappings between fields of the reconciliation staging data set and the OIM - User data set. Each mapping represents a reconciliation rule element. If there are child data sets, you must ensure that the names of fields of the reconciliation staging data set that are input fields for the matching-only mappings are not used in any of the reconciliation staging child data sets.

– To specify the key field for reconciliation matching, you create a matching-only mapping between the unique field of the reconciliation staging data set and the ID field of the OIM - Account data set. Along with the ID field, other fields of the OIM - Account data set can be (matching-only) mapped to corresponding fields of the reconciliation staging data set to create a composite key field for reconciliation matching.

Caution: If the name of a reconciliation staging field used in a matching-only mapping were to be reused as the name of a field in a reconciliation staging child data set, matching would not take place during a reconciliation run.

This known issue is explained in the Modify Connector Config Page section .

- **Create End-to-End Mapping**: If you are adding a field, select this check box if you want the same field to be added in all the data sets that are displayed to the right of the data set in which you are adding the field.
- **Multi-Valued Field**: Select this check box if you want to add a child data set. If you select this check box, the name that you specify in the Field Name field is used as the name of the child data set.

Note: If you select the Trusted Source Reconciliation check box on the Step 1: Provide Basic Information page, this check box (in selected or deselected state) is ignored. This is because the reconciliation of multivalued (child) data is not supported in trusted source reconciliation.

- **Data Type**: Select the data type of the field.

  After metadata detection, the String data type is applied by default to all the fields of the reconciliation staging and OIM - Account data sets. Where required, you must use the Data Type list to specify the actual data type of each field.
- **Length**: Specify the character length of the field.
- **Required**: Select this check box if you want to ensure that the field always contains a value.
- **Encrypted**: Select this check box if the value of the field must be stored in encrypted form in the Oracle Identity Manager database.
- **Password Field**: Select this check box if the value of the field must be encrypted on the process form. Values of fields for which this check box is selected are displayed as asterisks (*) on the process forms.

Note: If you select the Encrypted and Password Field check boxes, see "Password-Like Fields" in *Developing and Customizing Applications for Oracle Identity Manager* for information about guidelines that you must follow.

- **Lookup Field**: Select this check box if you want to make the field a lookup field.

3. Click **Continue**.
4. If you select the **Lookup Field** check box on the Step 1: Field Information page, the Step 2: Lookup Properties page is displayed. On this page, you can select and specify values for any combination of the lookup properties described in Table 9–4.

Table 9–4 Lookup Properties

Lookup Property	Value
Column Names	In the **Property Value** field, enter the name of the database column containing the values that must be displayed in the lookup window. If required, you can enter multiple database column names separated by commas. **Note:** If you select the Lookup Column Name property, you must also select the Column Names property, which is described later in this table. After you enter a value in the **Property Value** field, click **Submit**. The following SQL query can be used to illustrate how the Column Names and Lookup Column Name properties are used: `SELECT USR_FIRST_NAME, USR_LOGIN, USR_LAST_NAME FROM USR` Suppose you set the following as the values of the two properties: - Column Names: `USR_FIRST_NAME, USR_LAST_NAME` - Lookup Column Name: `USR_LOGIN` When the user selects a particular `USR_FIRST_NAME, USR_LAST_NAME` combination from the lookup window, the corresponding `USR_LOGIN` value is stored in the database.
Column Captions	In the **Property Value** field, enter the name of the column heading that must be displayed in the lookup window. If multiple columns are going to be displayed in the lookup window, enter multiple column captions separated by commas, for example, **Organization Name**, **Organization Status**. After you enter a value in the **Property Value** field, click **Submit**.
Column Widths	In the **Property Value** field, enter the character width of the column that must be displayed in the lookup window. This must be the same as the maximum length of the underlying field or column from which data values are drawn to populate the lookup field. If the lookup window is going to display multiple columns, enter multiple column widths separated by commas. After you enter a value in the **Property Value** field, click **Submit**.

Table 9–4 (Cont.) Lookup Properties

Lookup Property	Value

Lookup Query	To specify a value for the Lookup Query property:
	1. In the **Property Value** field, enter the SQL query (without the `WHERE` clause) that must be run when a user double-clicks the lookup field to populate the data columns displayed in the lookup window.
	2. Click **Submit**.
	3. On the Step 2: Add Validation page, select values from the following lists to create a `WHERE` clause for the `SELECT` statement that you specify in Step 1:
	- Filter Column
	- Source
	- Field Name
	From the values that you select, the `WHERE` clause is created as follows:
	`WHERE Filter_Column=Source.Field_Name`
	4. Click **Save**.
	To correctly display the data returned from a query, you must add a `lookupfield.header` property to the `xlWebAdmin_`*`locale`*`.properties` file.
	For example, consider the following SQL query:
	`SELECT usr_status FROM usr`
	To view the data returned from the query, you must add the following entry to the `xlWebAdmin_`*`locale`*`.properties` files:
	`lookupfield.header.users.status=User Status`
	If the `xlWebAdmin_locale.properties` file does not contain a `lookupfield.header` property for your specified query, the Identity System Administration displays a lookup window after you click the corresponding lookup icon.
	The syntax for a `lookupfield.header` property is as follows:
	`lookupfield.header.column_code=display value`
	The *`column_code`* portion of the entry must be lowercase and any spaces must be replaced by underscore characters (_).
	By default, the following entries for lookup field column headers are already available in the `xlWebAdmin_locale.properties` file:
	`lookupfield.header.lookup_definition.lookup_code_information .code_key=Value`
	`lookupfield.header.lookup_definition.lookup_code_information .decode=Description`
	`lookupfield.header.users.manager_login=User ID`
	`lookupfield.header.organizations.organization_name=Name`
	`lookupfield.header.it_resources.key=Key`
	`lookupfield.header.it_resources.name=Instance Name`
	`lookupfield.header.users.user_id=User ID`
	`lookupfield.header.users.last_name=Last Name`
	`lookupfield.header.users.first_name=First Name`
	`lookupfield.header.groups.group_name=Group Name`
	`lookupfield.header.objects.name=Resource Name`
	`lookupfield.header.access_policies.name=Access Policy Name`

Table 9–4 (Cont.) Lookup Properties

Lookup Property	Value
Lookup Code	In the **Property Value** field, enter the lookup definition code name. This code must generate all information pertaining to the lookup field, including lookup values and the text that is displayed with the lookup field when a lookup value is selected. The classification type of the lookup definition code must be of Lookup Type (that is, the Lookup Type option on the Lookup Definition form must be selected). To enter a lookup code, open the Lookup Definition form, query for the required code, and copy the code into the Property Value field. After you enter a value in the **Property Value** field, click **Submit**. **Note:** The Lookup Code property can be used to replace the combination of the **Column Captions**, **Column Names**, **Column Widths**, **Lookup Column Name**, and **Lookup Query** properties. In addition, the information contained in the Lookup Code property supersedes any values set in these five lookup properties. If you want to implement lookup fields reconciliation, create a scheduled task that populates the lookup code.
Lookup Column Name	In the **Property Value** field, enter the name of the database column containing the value that must be stored corresponding to the Column Names value selected by the user in the lookup window. If required, you can enter multiple database column names separated by commas. **Note:** If you select the Column Names property, you must also select the Lookup Column Name property. See the "Lookup Column Name" row in this table for more information about how these two properties are used. After you enter a value in the **Property Value** field, click **Submit**.
Auto Complete	If you enter `True` in the **Property Value** field, users can filter the values displayed in the lookup window by entering the first few characters of the value they want to select and double-clicking the lookup field. The outcome of this action is that only lookup values that begin with the characters entered by the users are displayed in the lookup window. For example, for the State lookup field, a user can enter `New` in the field. When the user double-clicks the State lookup field, only states that begin with `New` (for example, New Hampshire, New Jersey, New Mexico, and New York) are displayed in the lookup window. If you do not want to let users filter the display of values in the lookup field, enter `False` in the **Property Value** field. The default value of the **Auto Complete** property is `False`. After you enter a value in the **Property Value** field, click **Submit**.

If you want to edit the value of a property that is displayed in the table on the Step 2: Lookup Properties page, select the edit option for that property and click **Edit**. If you want to remove a property that is displayed in the table, select the delete option for that property and click **Delete**.

After you specify properties for the lookup field, click **Continue**.

5. If you select a transformation option from the Mapping Action list on the Step 1: Field Information page, the Step 3: Mapping page is displayed. Use this page to define the transformation function that you want to perform on the input data to the field that you are adding. The steps to be performed depend on the transformation provider option (concatenation, translation, or custom transformation provider) that you select on the previous page:

   If you select a predefined transformation provider (transformation, concatenation or translation), see Transformation Providers for detailed information about the procedure to specify parameter values for the predefined transformation provider.

That section also provides detailed information about configuring user account status reconciliation.

You must use the translation transformation provider if you want to configure the reconciliation of user account status information. This procedure is described in "Translation Transformation Provider" in *Developing and Customizing Applications for Oracle Identity Manager*.

After you specify values for the transformation provider, click **Continue**.

6. If required, select a validation check for the field and click **Add**. In other words, select the validation provider that you want to use.

   The validation options displayed in this list are based on the predefined validation Providers and any custom validation Providers that you create.

7. Click **Continue**, and click **Close**.

8. If you do not want to perform any other action on the Step 3: Modify Connector Configuration page, click the **Close** button that is displayed at the top of the page. You must perform the previous step before you click this Close button.

9.1.4.3.2 Removing Fields from Data Sets To remove a field from a data set:

1. Click the Delete icon for that field.

2. If you do not want to perform any other action on the Step 3: Modify Connector Configuration page, click the **Close** button that is displayed at the top of the page.

9.1.4.3.3 Removing Mappings Between Fields To remove a mapping:

1. Click the edit icon for the destination field of the mapping that you want to remove.

   Note: If the destination field itself is the source field for another mapping, that mapping is not removed.

2. On the Step 1: Field Information page, select **Remove Mapping** from the **Transformation** list.

3. Click **Continue**.

4. On the last page, click **Close**.

5. If you do not want to perform any other action on the Step 3: Modify Connector Configuration page, click the **Close** button that is displayed at the top of the page.

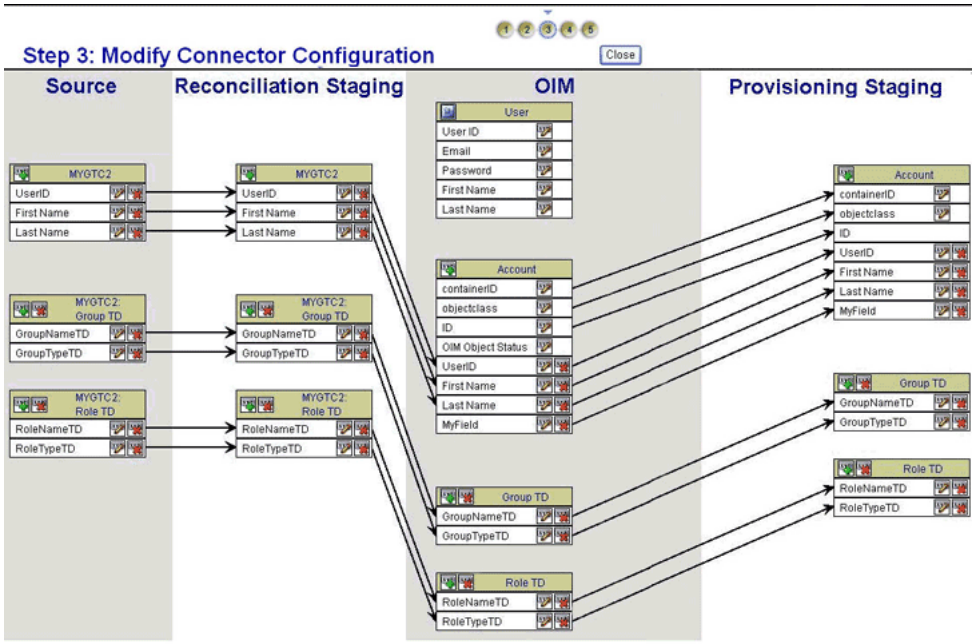
9.1.4.3.4 Removing Child Data Sets To remove a child data set:

1. Click the Delete icon for the data set.

2. If you do not want to perform any other action on the Step 3: Modify Connector Configuration page, click the **Close** button that is displayed at the top of the page.

Figure 9–2 shows the Step 3: Specify Connector Configuration page after the `MyField` field was added to the OIM - Account and provisioning staging data sets.

Figure 9–2 Step 3: Modify Connector Configuration Page After Addition of a Field



9.1.4.4 Step 4: Verify Connector Form Names Page

Use this page to specify form names for the process forms corresponding to the OIM - Account data set and its child data sets.

> **Note:** If you select the Trusted Source Reconciliation option on the Step 1: Provide Basic Information page, the OIM - Account data set and its child data sets are not created. Therefore, this page is not displayed if you select the Trusted Source Reconciliation option.

The generic technology connector framework automatically creates certain objects after you submit all the information required to create a generic technology connector. Parent and child process forms corresponding to the OIM - Account data sets are examples of objects that are automatically created. Each process form on a particular Oracle Identity Manager installation must have a unique name.

On the Step 4: Verify Connector Form Names page, the generic technology connector framework displays default names for these process forms based on the names of the corresponding data sets. You must verify and, if required, change the names of these forms to ensure that they are unique for this installation of Oracle Identity Manager. While changing the name of a form, you must use only ASCII characters. An error message is displayed if you specify non-unique form names or if any name contains non-ASCII characters.

> **Note:** You cannot revisit this page, so ensure that the form names that you specify meet all the requirements before you click **Continue**.

After you specify the form names, click **Continue**.

Instead of clicking Continue, you can click **Back** to return to the Step 2: Specify Parameter Values page. However, metadata detection does not take place if you make changes on this page and click the Continue button. This is to ensure that any customization in the data set structure and mappings made during the first pass through this page does not get overwritten. You can manually add or edit fields and mappings on the Step 3: Modify Connector Configuration page.

9.1.4.5 Step 5: Verify Connector Information Page

Use this page to review information that you have provided up to this point for creating generic technology connectors. The following is a page-wise explanation of the changes that are permitted on the earlier pages:

- Step 1: Provide Basic Information page

  You can use either the View link or Back button to reopen and view the information provided on the Step 1: Provide Basic Information page. You cannot change the information displayed on this page, because any change in this information would amount to creating a new generic technology connector.

- Step 2: Specify Parameter Values page

  You can use either the Change link or Back button to reopen this page. You can change parameter values on this page. However, metadata detection does not take place when you submit the changed values. This is to ensure that any customization in the data set structure and mappings made during the first pass through this page does not get overwritten. You can manually add or edit fields and mappings on the Step 3: Modify Connector Configuration page.

- Step 3: Modify Connector Configuration page

  You can use the Change link to reopen this page and add or edit fields and mappings.

- Step 4: Verify Connector Form Names page

  You cannot revisit this page.

After you verify all the information displayed on the Step 5: Verify Connector Information page, click **Create**.

At this stage, the generic technology connector framework creates all the standard connector objects on the basis of the information that you provide. The list of these objects includes the connector XML file, which is created and imported automatically into Oracle Identity Manager. Except for the form names, the names of the connector objects are in the *GTCname*_GTC format.

For example, if you specify DB_conn as the name of a generic technology connector that you create, all (except the forms) the connector objects are named DB_CONN_GTC.

At the end of the process, a message stating that the connector has been successfully created is displayed on the page.

Note: If the creation process fails, objects that are created are not automatically deleted.

9.1.5 Configuring Reconciliation

Note: If you select only the Provisioning option on the Step 1: Provide Basic Information page, you can skip this section because you need not configure reconciliation.

A reconciliation scheduled task is created automatically when you create the generic technology connector. To configure and run this scheduled task, follow the instructions in the "Creating and Managing Scheduled Tasks" in the *Oracle Fusion Middleware Administering Oracle Identity Manager*.

Note: The name of the scheduled task is in the following format:

*GTC_Name*_GTC

For example, if the name of the generic technology connector is WebConn, the name of the scheduled task is WebConn_GTC.

9.1.6 Configuring Provisioning

Note: If you select only the Reconciliation option on the Step 1: Provide Basic Information page, you can skip this section because you need not configure provisioning.

A process definition is one of the objects that are automatically created when you create a generic technology connector. The name of the process definition is in the following format:

*GTC_name*_GTC

For example, if the name of the generic technology connector is WebConn, the name of the process definition is WebConn_GTC.

The process tasks that constitute this process definition can be divided into two types:

- System-defined process tasks

  System-defined process tasks are included by default in all newly created process definitions.

- Provisioning-specific process tasks

  Provisioning-specific process tasks are included in the process definition of a generic technology connector only if you select the Provisioning option on the Step 1: Provide Basic Information page, regardless of whether or not you select the Reconciliation option.

The following are provisioning-specific process tasks:

- Create User
- Delete User
- Enable User
- Disable User

- Updated *Field_Name* (this task is created for each field of the OIM - Account data set, except the `ID` field)
- For mappings created between fields of the OIM - User data set and the provisioning staging data set, the following process tasks are created:
  - Change *User_data_set_field_name*
  - Edit *Provisioning_Staging_field_name*

  For example, suppose you create a mapping between the Last Name field of the OIM - User data set and the LName field of the provisioning staging data set. The following process tasks are automatically created along with the rest of the provisioning-specific process tasks:
  - Change Last Name
  - Edit LName

In addition, the following provisioning-specific process tasks are created for each child data set of the OIM - Account data set:

- Child Table *Child_Form_Name* row Inserted
- Child Table *Child_Form_Name* row Updated
- Child Table *Child_Form_Name* row Deleted

All provisioning-specific process tasks have the following default assignments:

- Target Type: `Group User With Highest Priority`
- Group: `SYSTEM ADMINISTRATORS`
- User: `XELSYSADM`

If required, you can modify these default assignments by following the instructions given in "Modifying Process Tasks" in *Developing and Customizing Applications for Oracle Identity Manager*.

9.1.7 Creating the Form and Publishing the Application Instance

To create the form and publish the application instance, which is created when you select both the provisioning and reconciliation options on the Step 1: Basic Information page, perform the following steps:

1. Create a form specific to the GTC resource object.
2. Attach the form to the GTC application instance.
3. Publish the GTC application instance to the required organizations.

Note: To view a provisioned account in the new UI, the process form should have a field for IT resource. The value for this IT resource field should be populated during a reconciliation run.

9.1.8 Enabling Logging

This is an optional step. Perform the procedure discussed in this section only if you want to enable logging for the generic technology connector.

See "Configuring Logging" on page 27-5 for information about enabling logging in Oracle Identity Manager.

9.2 Managing Generic Technology Connectors

The generic technology connector framework offers features that enable you to modify a generic technology connector. In addition, you can export or import a generic technology connector by using the Deployment Manager.

This section contains these topics:

- Modifying Generic Technology Connectors
- Exporting Generic Technology Connectors
- Importing Generic Technology Connectors

9.2.1 Modifying Generic Technology Connectors

Caution: The Design Console can be used to modify connector objects that are automatically created at the end of the generic technology connector creation process. If you use the Manage Generic Technology Connector feature to modify a generic technology connector whose connector objects have been customized by using the Design Console, all the customization work done using the Design Console would get overwritten. Therefore, Oracle recommends that you to follow one of the following guidelines:

- Do not use the Design Console to modify generic technology connector objects.

  The exception to this guideline is the IT resource. You can modify the parameters of the IT resource by using the Design Console. However, for the changes to take effect, you must purge the cache either before or after you modify IT resource parameters.

- If you use the Design Console to modify generic technology connector objects, do not use the Manage Generic Technology Connector feature to modify the generic technology connector.

In addition, you can modify only one connector at a time. If you try to use the Modify pages for two different connectors at the same time on the same computer, the Modify features would not work correctly.

To modify a generic technology connector:

1. Login to Identity System Administration.
2. Under Provisioning Configuration, click **Generic Connector**.
3. Search for the connector that you want to modify. To simplify your search, you can use a combination of the search criteria provided on this page. Alternatively, to view all the generic technology connectors that have been created on this Oracle Identity Manager installation, click **Search connectors** without specifying any search criteria.
4. In the results that are displayed, click the generic technology connector that you want to modify.
5. Click **Edit Parameters**. The Step 2: Specify Parameter Values page of the connector creation process is displayed. From this point onward, follow the procedure described in the Step 2 section.

Note: The only difference between this procedure and the procedure that you follow to create the generic technology connector procedure is that automatic metadata detection does not take place when you modify an existing generic technology connector.

Caution: If you modify attributes of fields of the OIM - Account data set or its child data sets, corresponding changes are not made in the Oracle Identity Manager database entries for these data sets. At the same time, no error message is displayed.

Therefore, Oracle recommends that you do not modify the fields or child data sets of the OIM - Account data set.

9.2.2 Exporting Generic Technology Connectors

You can export the XML file of a generic technology connector. This XML file contains definitions for all the objects that are part of the connector. If you want to use the same generic technology connector on a new Oracle Identity Manager installation, you must first export the XML file and import it into the new Oracle Identity Manager installation.

To export the connector XML file:

1. In the Oracle Identity Manager Advanced Administration, under System Management, click **Export Deployment Manager File**.
2. On the first page of the Deployment Manager Wizard, select **Generic Connector** from the list and click **Search**.
3. In the search results, select the generic technology connector whose XML file you want to export.
4. Click **Select Children**.
5. For the selected generic technology connector, select the child entities that you want to export and click **Select Dependencies**.
6. Select the dependencies that you want to export, and click **Confirmation**.
7. After you verify that the elements displayed on the page cover your export requirements, click **Add for Export**.
8. Click **Exit wizard and show full selection**, and click **OK**.

9.2.3 Importing Generic Technology Connectors

To copy a generic technology connector to a different Oracle Identity Manager installation:

1. If the connector uses custom providers, you must copy the files created during provider creation to the appropriate directories on the destination Oracle Identity Manager installation.
2. Export the connector XML file on the source Oracle Identity Manager installation.
3. Import the connector XML file on the destination Oracle Identity Manager installation.

Caution: You must ensure that the names you select for a generic technology connector and its constituent objects on a staging server do not cause naming conflicts with existing connectors and objects on the production server.

The following scenario explains why you must follow this guideline:

Suppose you create a generic technology connector on a staging server, and want to import the connector to a production server. While creating the generic technology connector on the staging server, you would have ensured that the names of the generic technology connector and the connector objects are unique on that server. At the same time, you must also ensure that the names are not the same as the names of connectors and connector objects on the production server.

If any of the names happen to be the same, the old objects would be overwritten by the new objects when you import the connector XML file from the staging server to the production server. No message is displayed during the overwrite process, and the process would lead to eventual failure of the affected connectors.

To ensure that you are able to revert to a working state in the event that an object is overwritten, you must create a backup of the destination Oracle Identity Manager database before you import a connector XML file.

To import the connector XML file:

1. In the Oracle Identity Manager Advanced Administration, under System Management, click **Import Deployment Management File**. A dialog box for locating files is displayed.
2. Locate and open the connector XML file from the directory into which you copy it.
3. Click **Add File**.
4. Click **Next**, **Next**, and **Skip**.
5. Click **View Selections**.

   The contents of the connector XML file are displayed on the Import page. You *may* see a cross-shaped icon along with some nodes. These nodes represent Oracle Identity Manager entities that are redundant. Before you import the connector XML file, you must remove these entities by right-clicking each node and selecting **Remove**.
6. Click **Import**. The connector file is imported into Oracle Identity Manager.

After you import the connector XML file, you must update the run-time parameters of the generic technology connector.

Note: These values are not copied in the connector XML file when you export it.

To update the values of the run-time parameters, follow the procedure described in "Modifying Generic Technology Connectors" on page 9-30.

10

Managing Application Instances

Application instance is a new abstraction used in 11*g* Release 2 (11.1.2.3.0). It is a combination of IT resource instance (target connectivity and connector configuration) and resource object (provisioning mechanism).

In pre-R2 releases, requests creation was based on name of resources and it was Administrator-centric, which needed good knowledge of technology. However in 11*g* Release 2 (11.1.2.3.0), accounts and entitlements of users are associated with application instances, and not with the IT resource instance or resource object. This makes it easier for an end user to operate.

Application instance is published to organizations and can be requested by users of those organizations. Suppose Microsoft Active Directory (AD) is to be provisioned to users across different organizations or departments across the world. You can define application instances consisting of the following:

- AD as the resource object
- Each AD server instance with the connectivity information, such as URL and password, as IT resources

This is because the resource object is same for all users, but the connectivity information, such as port number, can be different for users who are part of different organizations. Therefore, the AD resource object can be provisioned as an application instance without the user being aware of the connectivity information.

Application Instance is the provisionable entity. In order to get an account in a specific target, end users will need to request for the application instance. Instead of requesting for a resource and configuring IT resource instance separately, end user can request for an application instance. The request is subject to approval by an approver. When the request is approved, the resource is provisioned to the user, and an account is created in the target system.

> **Note:** If the request is coming from an authorizer, then it may not require approval, where as a request coming from an end user needs approval by approver.

This chapter describes application instances in the following topics:

- Developing Entitlements
- Managing Disconnected Resources

10.1 Application Instance Concepts

The application instance concepts are described in the following sections:

- Multiple Accounts Per Application Instance
- Entitlements
- Disconnected Application Instances
- Application Instance Security

10.1.1 Multiple Accounts Per Application Instance

Users in an enterprise can have multiple accounts in a single application instance. This is required in a scenario in which an HR administrator performs various tasks for other employees in the organization by using an administrative account. The same HR administrator logs in by using a separate user account when performing certain tasks for self. In this example, the same user requires two different accounts for logging in to the system and performs different types of operations.

In addition, supporting multiple accounts for users is required to prevent potential security threats. Suppose a user uses the same account for logging in to the environment, and performs administrative tasks, regular business tasks for self and others, and tasks related to IT infrastructure. If there is an intrusion in the system and the account is hacked, the hacker can access infrastructure data and other confidential information. If the user has multiple accounts for each type of task and the regular account is hacked, the confidential information related to IT infrastructure and other sensitive resources are secured from the hacker.

Oracle Identity Manager supports multiple accounts in a single application instance. The first account that is created is tagged as primary account, and there can be only one primary account for a user. The subsequent accounts created on the same application instance would be tagged as Other.

When the user gets provisioned to an application instance, the Oracle Identity Manager checks if it is the first account getting provisioned for the user in that application instance. If it is the first account, then the account is marked as primary. When existing user accounts are reconciled from application instances, the first account that gets reconciled is marked as primary. If the account marked as primary is not the actual primary account, then you can manually change the primary tag for the account and mark another account as primary.

10.1.2 Entitlements

An entitlement granted to an account on a target system enables the account owner (user) to perform a specific task or function. An entitlement can be a role, responsibility, or group membership. For example, if user Richard is granted the Inventory Analyst role on a target system, then Richard has a child process form that holds Inventory Analyst role data.

In Oracle Identity Manager, there is one process form for each account (resource) provisioned to an Oracle Identity Manager User. Entitlement data is stored in child process form. In the example described earlier, the process form for Richard's account on the target system has a child process form that holds Inventory Manager role data.

Note: To reconcile entitlements created in the target system into Oracle Identity Manager, you must first run the scheduled job for lookup field synchronization, and then run the Entitlement List scheduled job.

Attributes that constitute entitlement data stored on a child process form may vary from one target system to another. In addition, different types of entitlements, such as roles and responsibilities, may have different attributes.

Entitlements can be requested directly instead of first requesting a modify resource on user accounts. Entitlements are not part of the account data as the child forms are handled independently. A user can provision, modify, or revoke an entitlement. For the requested entitlements, the user can provide additional information that might help an approver during the approval process.

All types of entitlements are available for request in the request catalog. If the request for an administrative entitlement is approved, then it is associated to the primary account. In addition, the requester can select target accounts, and approvers can also modify the target account.

You can edit the entitlements by using the Application Instances section of the Oracle Identity System Administration.

See "Developing Entitlements" on page 10-17 for detailed information about entitlements.

10.1.3 Disconnected Application Instances

You might deploy self service, delegated administration, request management, and role-based provisioning features in Oracle Identity Manager, and might not deploy provisioning and reconciliation connectors to automate provisioning. After completion of delegated administration operation, request-approval, or role-based provisioning, a manual provisioning task is assigned to an administrator. The administrator then manually performs the provisioning in the target application instance. An example of this is provisioning of an access card, which is physical. Because Oracle Identity Manager cannot provision a physical access card, the application instance of the disconnected resource is to be provisioned.

To achieve provisioning of disconnected resource, you can create application instances of the disconnected type. The manual provisioning administrator can use the Inbox section of the Oracle Identity Self Service to update all fields in the request. After the manual provisioning administrator submits the manual provisioning worklist item, the provisioning infrastructure marks the underlying provisioning task to be completed based on the response of the manual provisioning administrator. If the administrator specifies that task is manually completed, then the status is changed to provisioned.

10.1.4 Application Instance Security

The Application Instance is also the entity with which security primitives are associated via the organization publishing mechanism. Only those organizations that have the application instance published to them are able to provision to the targets.

10.2 Managing Application Instances

You manage application instances by using Oracle Identity System Administration. This includes:

- Creating Application Instances
- Searching Application Instances
- Modifying Application Instances
- Deleting Application Instances
- Creating and Modifying Forms

> **See Also:** "Converting a Disconnected Application Instance to Connected" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about converting a disconnected application instance to a connected application instance

10.2.1 Creating Application Instances

To create an application instance:

1. Login to Oracle Identity System Administration.
2. In the left pane, under Provisioning Configuration, click **Application Instances**. The Application Instances page is displayed.
3. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Application Instance page is displayed.
4. Enter the values of the attributes, as listed in Table 10–1:

Table 10–1 Fields in the Create Application Instance Page

Attribute	Description
Name	The name of the application instance. This is a required field. **Note:** If you enter non-ASCII characters in the Name field, then an error message is displayed when you try to save the application instance. It is recommended that you enter only ASCII or alphanumeric characters in the Name field.
Display Name	The display name of the application instance. This is a required field.
Description	A description of the application instance.
Disconnected	Select if you want to specify the application instance as disconnected. Selecting this option creates a new approval process that is assigned to the manual provisioning administrator. See "Disconnected Application Instances" on page 10-3 for more information. **Note**: Disconnected application instance can only be created when a sandbox is active. See "Managing Sandboxes" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for more information about sandbox.
Resource Object	The resource object name. You can click the search icon next to this field to search and select a resource object.
IT Resource Instance	The IT resource instance name. You can click the search icon next to this field to search and select an IT resource instance.

Table 10–1 (Cont.) Fields in the Create Application Instance Page

Attribute	Description
Form	Select the form or dataset name. The forms associated with the selected resource object are populated in the Forms list. Here, only pre-existing forms can be selected.
Parent AppInstance	The application instance name that you want to specify as a parent to the new application instance. The new application instance inherits all the properties of the parent application instance. Resource must be assigned as 'Depends on' in the Design Console to populate this lookup.

5. Click **Save**. The application instance is created, and the details of the application instance is displayed in a page.

10.2.2 Searching Application Instances

To search for application instances:

1. In the Oracle Identity System Administration, under Provisioning Configuration, click **Application Instances**. The Application Instances page is displayed.
2. Select any one of the following:
   - **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 searchable application instance attribute fields, such as Display Name, specify a value.

   For some attributes, select the attribute value from the lookup. For example, to search all application instances with a particular resource object, specify the resource object name in the Resource Object field.
4. For each attribute value that you specify, select a search operator from the list. The following search operators are available:
   - Starts with
   - Ends with
   - Equals
   - Does not equal
   - Contains
5. To add a searchable application instance attribute to the Application Instances page, click **Add Fields**, and select the attribute from the list of attributes.

   For example, if you want to search all application instances under a parent application instance, then you can add the Parent AppInstance attribute as a searchable field and specify a search condition.
6. Optionally click **Reset** to reset the search conditions that you specified. Typically, you perform this step to remove the specified search conditions and specify a new search condition.

7. Click **Search**. The search result is displayed in a tabular format.

   Tip: You can use the Query By Example feature to refine your search based on specific values. For more information, see "Query By Example" in *Performing Self Service Tasks with Oracle Identity Manager*.

10.2.3 Modifying Application Instances

You can open an application instance and modify the attributes, assign and revoke organizations to which the application instance is available, and edit the entitlements associated with the application instance. These tasks are described in the following sections:

- Modifying Application Instance Attributes
- Managing Organizations Associated With Application Instances
- Managing Entitlements Associated With Application Instances

10.2.3.1 Modifying Application Instance Attributes

To modify the attributes of an application instance:

1. In the Application Instances page, search and select the application instance that you want to open.
2. From the Actions menu, click Open. Alternatively, click Open on the toolbar. You can also click the Display Name of the application instance.

   The Application Instance details page is displayed.
3. Ensure that the Attributes tab is displayed. The fields that you are not allowed to modify are grayed out.
4. Edit the values in the fields, such as Display Name, Description, Form, and Parent AppInstance.
5. Click **Apply**. The attribute modifications are saved.
6. Run the Catalog Synchronization Job scheduled job.

   Note: The Catalog Synchronization Job should be run preferably in Incremental mode so that changes, such as add, update, and delete, in base entity application instance and entitlements are synced to catalog DB.

10.2.3.2 Managing Organizations Associated With Application Instances

You must make an application instance available for requesting and subsequent provisioning to users by publishing the application instance to an organization. The users in that organization or the users who has User Viewer role in that organization or the users who has Application Instance Viewer role + User Viewer Role in that organization can request for application instance. For information about authorization in Oracle Identity Manager, see *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

In the Organizations tab of the Application Instance details page, you can publish the application instance to organizations, and revoke organizations from the application instance.

In addition, you can publish the application instance to an organization and its suborganizations so that users of the suborganizations can also request for the application instance. You can also publish an application instance to organizations with entitlements so that users of the organization can request for the application instance with the entitlements associated with it.

> **Note:** An administrator user can publish an entity to any organization that the administrator can view. For example, an Entitlement Administrator can publish entitlements with administrative permissions to any organization on which the Entitlement Administrator has view permission.

This section describes the following tasks:

- Publishing an Application Instance to Organizations
- Revoking Organizations From an Application Instance

10.2.3.2.1 Publishing an Application Instance to Organizations

To publish an application instance to organizations:

1. In the Application Instance details page, click the **Organizations** tab. A list of organizations to which the open application instance is published is displayed.

   For each organization, the **include sub-orgs** option is displayed in the Hierarchy Aware column. Select this option to make the open application instance available to the organization and its suborganizations. Deselect this option to make the open application instance available to the organization only.

2. From the Actions menu, click **Assign**. Alternatively, click **Assign** on the toolbar. The Select Organizations dialog box is displayed.

3. Search for the organizations to which you want to publish to the open application instance.

   > **Note:** If you are using Oracle Identity System Administration in French on Google Chrome web browser, the right arrow may be missing or truncated in the search panel of the Select Organizations dialog box. To fix this issue, verify the display language setting in Chrome and change it to French if necessary.

4. Click **Add Selected**. The selected organizations are added to the Selected Organizations table.

   If you want the select all organizations, then click **Add All**.

5. For each organization added to the Selected Organizations table, a checkbox is displayed in the Hierarchy column. Select the **Hierarchy** option to publish the open application instance to the suborganizations of the selected organization.

   To publish the open application instance to the selected organizations only, leave the **Hierarchy** option deselected.

6. Select the **Apply to Entitlement** option to publish the open application instance to the selected organizations with the entitlements associated with the application instance. Otherwise, leave this option deselected.

7. Click **Select**. The application instance is published to the selected organizations.

The **include sub-orgs** option is displayed for the organizations for which you selected the **Hierarchy** option in the Select Organizations dialog box.

10.2.3.2.2 Revoking Organizations From an Application Instance

To revoke an organization from an application instance:

1. In the Organizations tab, select an organization that you want to revoke from the open application instance.
2. From the Action menu, select **Revoke**. Alternatively, click **Revoke** on the toolbar. A confirmation box is displayed with the selected organization.
3. Click **Yes** to confirm. The organization is revoked from the application instance.

> **Tip:** To revoke from suborganization of the organization to which the application instance is published, deselect the corresponding **include sub-orgs** option, and click **Apply**.

10.2.3.3 Managing Entitlements Associated With Application Instances

You modify the entitlements associated with application instances to change the entitlement attribute values, publish or revoke the entitlements to organizations, as described in the following sections:

- Modifying Entitlement Attributes
- Publishing an Entitlement to an Organization
- Revoking an Entitlement from an Organization

10.2.3.3.1 Modifying Entitlement Attributes

To modify the attributes of an entitlement associated with an application instance:

1. In the Application Instance details page, click the **Entitlements** tab. A list of entitlements associated with the open application instance is displayed.
2. Select the entitlement that you want to modify.
3. From the Actions menu, select **Edit**. Alternatively, click **Edit** on the toolbar. The details of the selected entitlement is displayed in a page.
4. Change the values of the attributes, such as Display Name and Description, and click **Save**. The entitlement modifications are saved.
5. Run the Catalog Synchronization Job scheduled job.

10.2.3.3.2 Publishing an Entitlement to an Organization

To publish an entitlement associated with an application instance to an organization:

1. In the Application Instance details page, click the **Entitlements** tab. A list of entitlements associated with the open application instance is displayed.
2. Select the entitlement that you want to publish. The entitlement details is displayed at the bottom of the page.
3. From the Actions menu, select **Assign**. Alternatively, click **Assign** on the toolbar. The Select Organizations dialog box is displayed.
4. Search and select the organization to which you want to publish the entitlement.
5. Click **Add Selected**. The organization is added to the Selected Organizations list.

   If you want to publish the entitlement to all organizations, then click **Add All**.

6. Optionally, select the **Hierarchy** option if you want to publish the entitlement to the suborganizations of the selected organization.
7. Click **Select**.
8. Run the Catalog Synchronization Job scheduled job.

10.2.3.3.3 Revoking an Entitlement from an Organization

To revoke an entitlement associated to an application instance from an organization:

1. In the Application Instance details page, click the **Entitlements** tab. A list of entitlements associated with the open application instance is displayed.
2. Select the entitlement that you want to revoke. The entitlement details is displayed at the bottom of the page.
3. If you want to revoke the entitlement from the suborganizations of the organization, then keep the **include sub-orgs** option selected.
4. From the Actions menu, select **Revoke**. Alternatively, click **Revoke** on the toolbar. A warning is displayed asking for confirmation.
5. Click **Yes**.
6. Run the Catalog Synchronization Job scheduled job.

10.2.4 Deleting Application Instances

An application instance can be deleted in any one of the following ways:

- Deleting the application instance from the Application Instances section of the Oracle Identity System Administration.
- Deleting the IT resource, which is a constituent of the application instance.

When you delete an application instance by using any one these methods, the application instance is not hard-deleted from Oracle Identity Manager. The application instance is soft-deleted. This is because accounts provisioned as a result of the application instance might exist in the target system. Therefore, after deleting an application instance, you must run a scheduled job to achieve the following:

- Unpublish the application instance from the entity publication
- Unpublish the associated entitlements from the entity publication
- Revoke, or hard-delete, or mark as deleted all the accounts for the application instance

To delete an application instance:

1. In Oracle Identity System Administration, under Provisioning Configuration, click **Application Instances**. The Application Instances page is displayed with a list of application instances that are published to your organization.
2. Search and select the application instance that you want to delete.
3. From the Actions menu, select **Delete**. Alternatively, click **Delete** on the toolbar. A message box is displayed asking for confirmation.
4. Click **Delete** to confirm. The application instance is soft-deleted in Oracle Identity Manager.

   You can also delete an application instance by deleting the IT resource of the application instance. For information about deleting IT resources, see "Managing

IT Resources" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

5. Run the Application Instance Post Delete Processing Job scheduled job. This scheduled job can be run in any one of the following modes:

   - **Revoke:** This mode is used when the application instance is deleted, but the provisioned accounts in the target system still exist. Using the Revoke mode deletes the accounts from the target system.
   - **Delete:** This mode is used when the target system no longer exists, and there are no traces of the accounts in Oracle Identity Manager. Using the Delete mode hard-deletes the accounts from all provisioning tasks and targets, and subsequently from Oracle Identity Manager. If this mode is used when the accounts are assigned in Oracle Identity Manager, then the accounts are listed as deprovisioned under the users they are assigned to.
   - **Decommission:** This mode is used when the target system no longer exists and the provisioned accounts cannot be revoked from the target system. Using the Decommission mode changes the account status to Revoke without keeping the accounts in Oracle Identity Manager in provisioned state.

   For information about scheduled jobs, see "Managing the Scheduler" on page 18-1.

   Note: The Application Instance Post Delete Processing Job scheduled job can be run after deleting each application instance.

6. Run the Catalog Synchronization Job scheduled job. This scheduled job identifies the soft-deleted application instances, and removes them from the catalog.

   Note:

   - The Catalog Synchronization Job scheduled job run is independent of the Application Instance Post Delete Processing Job run. This means that the Catalog Synchronization Job scheduled job removes the soft-deleted application instances from the catalog even if Application Instance Post Delete Processing Job is not run after soft-deleting the application instances.
   - Catalog Synchronization Job should be run preferably in Incremental mode so that changes, such as add, update, and delete, in base entity application instance and entitlements are synced to catalog DB.

10.2.5 Creating and Modifying Forms

In the Application Instances section of Oracle Identity System Administration, you can create and modify forms associated with the resource objects, and subsequently with the application instances.

See Also:

- See "Managing Sandboxes" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about sandbox.
- See Chapter 6, "Managing Forms" for information about creating forms.
- See Chapter 7, "Configuring Custom Attributes" for information about configuring custom attributes.

This section describes the following topics:

- Creating Forms Associated With Application Instances
- Modifying Forms Associated With Application Instances
- Localizing Application Instance Form

10.2.5.1 Creating Forms Associated With Application Instances

To create a form associated with an application instance:

Note: You cannot create forms directly. Before creating forms, you must create a sandbox and activate it. See "Managing Sandboxes" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about creating and activating a sandbox.

1. Login to Oracle Identity System Administration.
2. Create and activate a sandbox. A warning message is displayed if no sandbox is activated. For detailed instructions on creating and activating a sandbox, see the "Managing Sandboxes" section of *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager.*
3. In the left pane, under Provisioning Configuration, click **Form Designer**. The Form Designer page is displayed.
4. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Form page is displayed.
5. In the Resource Type field, specify a resource object with which you want to associate the form. To do so:
   a. Click the lookup icon next to the Name field. The Search and Select: Name dialog box is displayed.
   b. In the Name field, enter the name of the resource object you want to search. You can leave this field blank if you want to display all resource objects.
   c. Click **Search**. The resource objects that match the search condition are displayed.
   d. Select the resource object that you want to associate with the form, and click **OK.** The resource object name is displayed in the Name field of the Create Form page.
6. In the Form Name field, enter a form name.
7. (Optional) Select any one of the available options for Form Type:

- **Parent Form + Child Tables (Master/Detail)**
- **Parent Form (Master)**
- **Parent Form + Child Tables for Non Entitlement (Master/Detail)**

8. (Optional) Select the **Generate Entitlement Forms** option if you want to associate the new form with the entitlements. Using this form, users can provide additional information that might help an approver during the approval process.
9. In the Available form fields section, a list of form field names along with description and Display Name are displayed. These fields are available for the form you are creating. For each available form field, you can select the Bulk Update option. Selecting this option makes the form field available for updating the entities in bulk.
10. In the Create Application Instance page or the Attributes tab of the Application Instance details page, click **Refresh** adjacent to the Form field.
11. Select the newly created form in the Form list and click **Apply.**

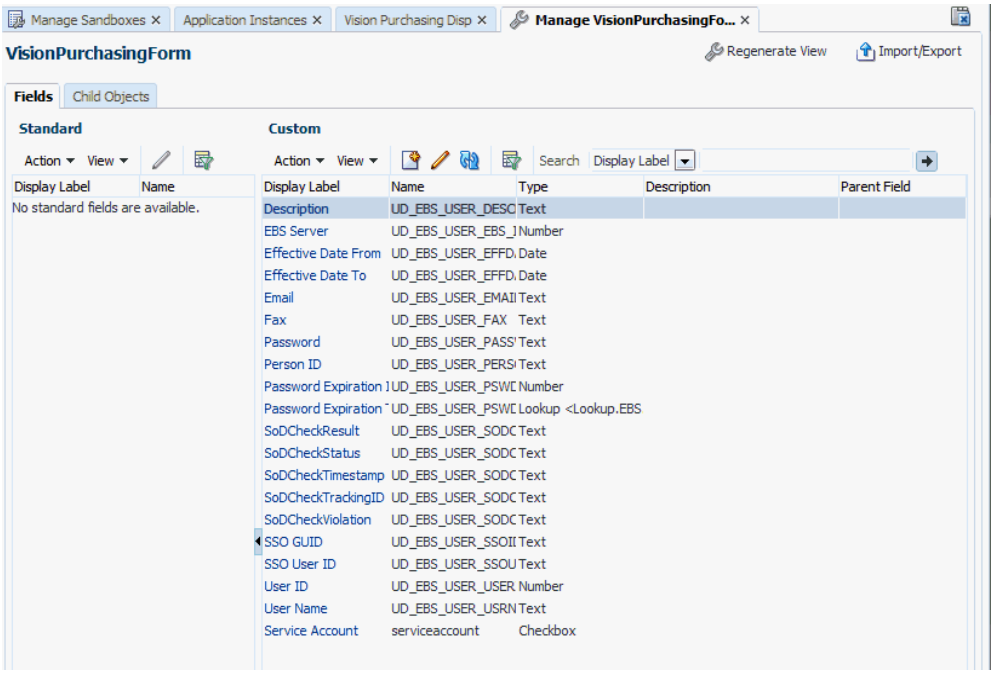
10.2.5.2 Modifying Forms Associated With Application Instances

Note: You cannot modify forms directly. Before creating forms, you must create a sandbox and activate it. See "Managing Sandboxes" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about modifying and activating a sandbox.

To modify a form associated with an application instance:

1. Open the Create Application Instance page or the Attributes tab of the Application Instance details page.
2. From the Form list, select the form you want to modify.
3. Click **Edit** to right of the Form field. The Manage Form page is displayed, as shown in Figure 10–1:

Figure 10–1 The Manage Form Page



For detailed information about modifying forms, see "Developing Process Forms" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

For information about creating and editing custom fields, see "Configuring Custom Attributes" on page 7-1.

4. (Optional) If you want to associate a form with an entitlement, then you can regenerate the form to allow users to provide additional information that might help the approver during the approval process. To do so, click **Regenerate View**. In the Regenerate View popup window, select the **Generate Entitlement Forms** checkbox. See "Modifying Forms By Using the Form Designer" on page 6-3 for information about the options available in the Regenerate View popup window.

> **Note:** If you have upgraded Oracle Identity Manager to release 11.1.2.2.0, then you must regenerate all the forms to use this feature.

10.2.5.3 Localizing Application Instance Form

To localize the application instance form:

1. Create an application instance of connector with a form attached to it.
2. Login to Oracle Enterprise Manager.
3. Go to **Application Deployments**, **oracle.iam.console.identity.sysadmin.ear**, **MDS Configuration**.
4. Click **Export** and save the archive to the host.
5. Unzip the archive, and open the *SAVE_LOCATION*\xliffBundles\oracle\iam\ui\runtime\BizEditorBundle.xlf file in a text editor.

Note: This file may not exist in MDS. If it does not exist, then create a new one, but the path must be the same.

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

   a. Search and replace the following:

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

   With the following for Japanese language:

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

   b. Search for the application instance code. This procedure shows a sample edit for JDE application instance. The original code is:

   ```
   <trans-unit
   id="${adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceB
   undle']['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.u
   serEO.UD_JDE_LANGUAGE__c_description']}">
   <source>Language</source>
   </target>
   </trans-unit>
   <trans-unit
   id="sessiondef.oracle.iam.ui.runtime.form.model.JDEArj.entity.JDEArjEO.UD_
   JDE_LANGUAGE__c_LABEL">
   <source>Language</source>
   </target>
   </trans-unit>
   ```

   c. Open the resource file from the connector package, for example JDEdwards_ja.properties, and get the value of the attribute from the file, for example, global.udf.UD_JDE_LANGUAGE=\u8A00\u8A9E.

   d. Replace the original code shown in step 6b with the following:

   ```
   <trans-unit
   id="${adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceB
   undle']['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.u
   serEO.UD_JDE_LANGUAGE__c_description']}">
   <source>Language</source>
   <target>\u8A00\u8A9E</target>
   </trans-unit>
   <trans-unit
   id="sessiondef.oracle.iam.ui.runtime.form.model.JDEArj.entity.JDEArjEO.UD_
   JDE_LANGUAGE__c_LABEL">
   <source>Language</source>
   <target>\u8A00\u8A9E</target>
   </trans-unit>
   ```

   e. Repeat steps 6a through 6d for all attributes of the process form.

   f. Save the file as BizEditorBundle_ja.xlf.

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

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

8. Logout or Oracle Identity Manager and login again.

10.3 Configuring Application Instances

You can configure application instances by using Oracle Identity System Administration. This includes:

- Section 10.3.1, "Configuring an Resource Object"
- Section 10.3.2, "Configuring IT Resource"
- Section 10.3.3, "Configuring Password Policies for Application Instances"

10.3.1 Configuring an Resource Object

For information about configuring a resource object, see "Resource Objects Form" in the *Developing and Customizing Applications for Oracle Identity Manager* at the following URL:

`https://docs.oracle.com/cd/E40329_01/dev.1112/e27150/resmgt.htm#OMDEV2468`

10.3.2 Configuring IT Resource

For information about configuring an IT resource, see "Creating IT Resources" on page 8-1 and "Managing IT Resources" on page 8-3.

An application instance can be configured for only one IT resource. If the process form requires value of two or more IT resources for provisioning an account, then it cannot be configured directly from the UI. To configure two or more IT resources for provisioning an account:

1. Identify the main IT resource for the account and configure the application instance with that.
2. Use entity adapter to populate the value for other required IT resources. For example, the Microsoft Exchange connector 9.1.1.7 requires an IT resource value of AD IT resource and Exchange IT resource to provision Exchange account. Here are the steps to make it work in R2
   a. Create an application instance with Exchange IT resource, and choose AD application instance as parent application because it is a dependent resource for Exchange.
   b. Configure an entity adapter to pass the value of the AD IT resource to the process form. To do so:

      i.) Keep a track of the dependent IT resource name, such as Exchange, and independent IT resource name, such as AD. This can be in the code, or externalized in a lookup and then initialized in the code.

      ii.) Create an entity adapter that takes long as a parameter. This is the parent IT resource key that is populated.

      iii.) In the adapter code, find the parent IT resource name and do a reverse lookup on the child IT resource name by using the map mentioned in step i.

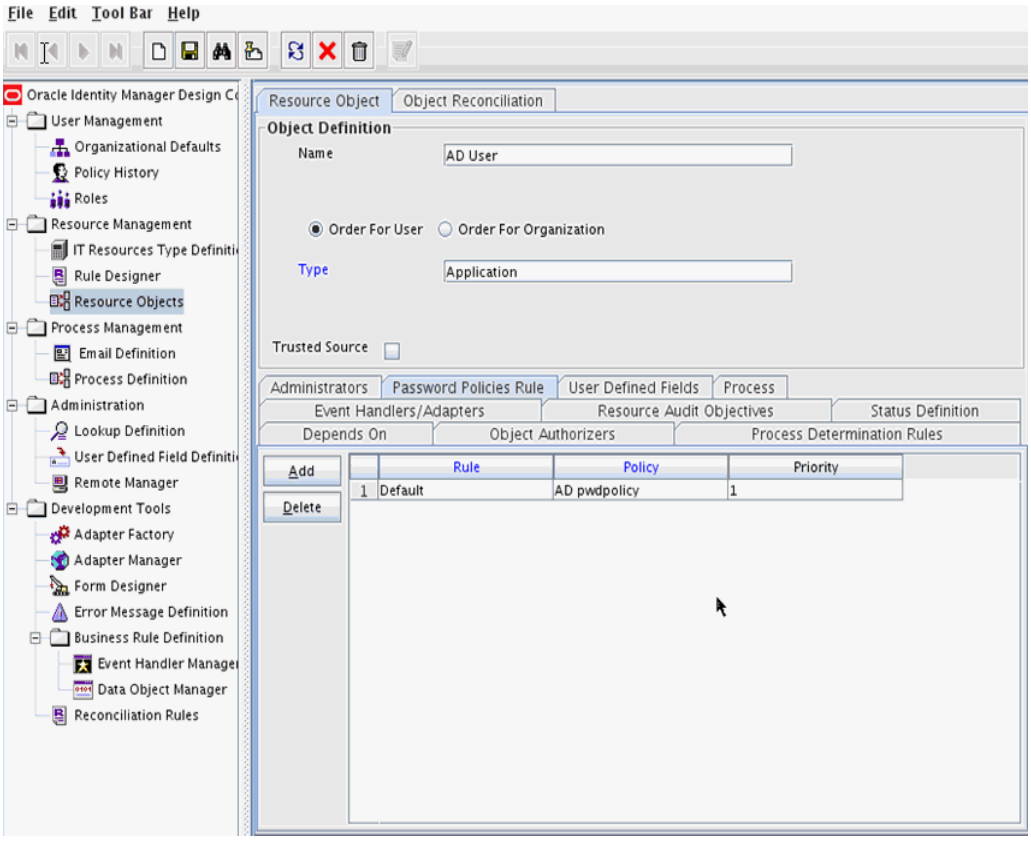
iv.) From the child IT resource name, get the child IT resource key as a long and return it. The entity adapter return value gets set on the child IT resource field on the process form.

10.3.3 Configuring Password Policies for Application Instances

Perform the following steps to configure the password policy for application instances:

1. Login to Oracle Identity Self Service.
2. Create a password policy for the application instance by setting a new rule for the password. See "Managing Password Policies" in *Performing Self Service Tasks with Oracle Identity Manager* for information about creating and managing password policies.
3. After you set the password policy for an Application Instance, you need to attach the new policy to the connected (AD User) application instance. To do so:
   a. Go to Design Console.
   b. Under Resource Management, click **Resource Objects**.
   c. Click on Password Policies Rule tab.
   d. Select the new password policy (AD pwdpolicy) that you created to attach it to the connected application instance.
   e. Click **Add**.

Figure 10–2 Attach Password Policy to Application Instance



10.4 Developing Entitlements

An entitlement granted to an account on a target system enables the account owner (user) to perform a specific task or function. An entitlement can be a role, responsibility, or group membership. For example, if user Richard is granted the Inventory Analyst role on a target system, then Richard can use that entitlement to access and generate inventory-related reports from the target system.

In Oracle Identity Manager, there is one process form for each account (resource) provisioned to an Oracle Identity Manager User. Entitlement data is stored in child process forms of the process form. In the example described earlier, the process form for Richard's account on the target system has a child process form that holds Inventory Manager role data.

Entitlements can be requested directly instead of first requesting a modify resource on user accounts. Entitlements are not part of the account data as the child forms are handled independently. A user can provision, modify, or revoke an entitlement. For the requested entitlements, the user can provide additional information that might help an approver during the approval process.

Attributes that constitute entitlement data stored on a child process form may vary from one target system to another. In addition, different types of entitlements, such as roles and responsibilities, may have different attributes. For example, Target System A contains the following role data attributes:

- Role Name
- Role Description
- Start Date
- End Date

The same target system can have a different set of attributes for responsibility data:

- Responsibility ID
- Date Assigned
- Proxy User
- Escalation User

You can mark or highlight the attribute that uniquely identifies an entitlement on a target system. For the sample role and responsibility data attributes listed earlier, the Role Name and Responsibility ID attributes uniquely identify the role and responsibility entitlements on Target System A. By marking attributes that uniquely identify entitlements, you enable the capture of entitlement data that can be used by other identity management solutions and also displayed in reports.

Note: If you are using the SAP User Management connector release 9.*x* with this release of Oracle Identity Manager, then perform the following steps for the Roles and Profiles entitlements to work correctly:

1. In the Role Child Form, from the Role System Name field, remove the Entitlement and Required properties.
2. In the Profiles Child Form, from the Profile System Name field, remove the Entitlement and Required properties.

This section discusses the following sections:

10.4.1 Available Entitlements and Assigned Entitlements

A target system can have a set of entitlements defined and ready for assignment to accounts (users) on the target system. When you integrate this target system with Oracle Identity Manager, you can import (synchronize) entitlement data from the target system into the LKV table on Oracle Identity Manager.

Note: If you use a predefined connector to integrate the target system, then you can use scheduled tasks to fetch entitlement data into this table.

The Entitlement List scheduled job is run synchronize the entitlements to the request catalog. An entitlement is available when it can be found in the request catalog. See Section 13.4.2.3, "Ongoing Synchronization" for more information about configuring Catalog Synchronization.

During a provisioning operation, you request the entitlement through the Catalog. You can also populate the entitlement data along with the parent data as request data set when submitting a request for an application instance.

In this guide, entitlements assigned to accounts are called assigned entitlements. Data about assigned entitlements is stored in child process form tables.

10.4.2 Entitlement Data Capture Process

After you mark the entitlement attribute in each child process form, capture of data about available entitlements take place.

The following steps describe how data about available entitlements is captured:

Note:

- You must mark the entitlement attribute in each child process form to enable the process described in these steps. The procedure is described later in this chapter.
- Make sure that the parent form has the latest child form version. It does not automatically happen when you create, edit, and activate the child parent without doing the same with the parent form. The Entitlement field can be marked from the Form Designer, which takes care of activating the parent/child forms.

1. Data about available entitlements is stored in the LKV table through synchronization with the target system.
2. You schedule and run the Entitlement List scheduled task.
3. The schedule task identifies the entitlement through the entitlement property in process form.
4. The scheduled task copies data about available entitlements from the LKV table to the ENT_LIST table.

10.4.3 Marking Entitlement Attributes on Child Process Forms

You must mark the entitlement attribute in the child process form UD_ table for resources for which you want to capture entitlement data. Suppose there are 15 target systems in your operating environment. If you want to capture entitlement data from 12 of 15 resources, then you must mark the entitlement attribute in those 12 resources.

Apply the following guidelines while performing the procedure described in this section:

- On a child process form, only one attribute holding entitlement data can be marked.
- The attribute that you mark must be of the LookupField type and its property must be one of the following:
  - Lookup code
  - Lookup query

    The Lookup query must satisfy the following conditions:

    * The query uses the LKU and LKV tables
    * The Lookup code in the query is from the LKU table
    * The LKV_ENCODED column value is used for saving
    * The LKV_DECODED column value is used for display purposes

To mark a field as an entitlement in a child process form:

1. Login to Identity System Administration.
2. Using the Form Designer, create a child form attribute, as described in "Creating a Custom Child Form Attribute" on page 7-6. Make sure that the **Entitlement** and **Searchable** options are selected when creating an attribute for entitlement.

10.4.4 Duplicate Validation for Entitlements or Child Data

Oracle Identity Manager validates duplicate entitlement or child data based on the following attributes, which ever is set:

- Key attribute
- Entitlement attribute

The configuration of the above mentioned attributes are checked prior to validating duplicates in the child data. Table 10–2 summarizes the possible valid and invalid configurations.

Table 10–2 Possible Scenarios and Duplicate Validation Basis

		Configuration Validation	
Entitlement Attribute	**Key Attribute for Reconciliation Field Mapping**	**Connected Application Instance**	**Disconnected Application Instance**
Not defined **Note**: In this scenario, the user is at a risk of adding duplicate entitlements or child data as the configurations are not defined properly. A warning message is logged on the server asking the user to define entitlement attribute and matching reconciliation field mapping.	Not defined	Valid	Valid
Defined. One attribute, say UD_CHILD1_ENT1 has Entitlement=true **Note**: Entitlement attribute does not have a matching key attribute defined in reconciliation field mapping.	Not defined	Invalid	Valid
Not defined	Defined. One attribute, say UD_CHILD1_ENT1 is set as the key attribute in recon field mapping.	Valid	Valid

Table 10–2 (Cont.) Possible Scenarios and Duplicate Validation Basis

		Configuration Validation	
Entitlement Attribute	**Key Attribute for Reconciliation Field Mapping**	**Connected Application Instance**	**Disconnected Application Instance**
Defined. One attribute, say UD_CHILD1_ENT1 has Entitlement=true	Defined. One attribute, say UD_CHILD1_ENT1 is set as the key attribute in recon field mapping.	Valid	Valid
Defined. One attribute, say UD_CHILD1_ENT1 has Entitlement=true **Note**: Entitlement attribute is a subset of the reconciliation field mapping key attributes.	Defined. Two or more attributes, say UD_CHILD1_ENT1 and UD_CHILD1_ENT2 are defined as key attributes in recon field mapping for child table UD_CHILD1.	Valid	Valid
Defined. One attribute, say UD_CHILD1_ENT1 has Entitlement=true **Note**: Entitlement attribute does not have a matching key attribute defined in reconciliation field mapping.	Defined. One or more attributes, say UD_CHILD1_ENT2 and UD_CHILD1_ENT3 are defined as key attributes in recon field mapping	Invalid	Invalid

Oracle recommends configuring both the entitlement attribute and the matching key attribute for the child data in reconciliation field mappings to enable effective validation.

Once a valid configuration is detected, duplicates are validated based on the operation as listed in Table 10–3.

Table 10–3 Duplicate Validation Based on Operation

Operation	Duplicate Validation Description
Adding entitlement(s)	The attribute for which "Entitlement=true" property is defined.
Adding child data	The attribute that is the key attribute in the reconciliation field mappings.

Note: Oracle recommends configuring both the entitlement attribute and the key attribute for the child data in reconciliation field mappings to enable effective duplicate entitlement or child data validation.

10.4.5 Configuring Scheduled Tasks for Working with Entitlement Data

You configure the following scheduled tasks for working with entitlement data:

- Entitlement List
- Entitlement Assignments

10.4.5.1 Entitlement List

The Entitlement List scheduled task identifies the entitlement attribute from the child process form table and then copies entitlement data from the LKV table into the ENT_LIST table. A record created in the ENT_LIST table corresponds to an entitlement defined on a particular target system.

You must set a schedule for this task depending on how frequently new entitlements are defined on the target systems in your operating environment. In addition, you must run this scheduled task when new target systems are integrated with Oracle Identity Manager. In other words, you must run this task each time you mark a new entitlement. After the connector scheduled tasks fetch lookup field data from the target system into the LKV table, you can run the Entitlement List scheduled task to copy that entitlement data into the ENT_LIST table.

This scheduled task also handles updates to or deletion of entitlements from the target system. For example, if the Senior Accounts Analyst role is removed from the target system, then the connector scheduled task removes the entry for that role from the LKV table. When the Entitlement List scheduled task is run, it marks the row containing the role in the ENT_LIST table as a deleted row.

10.4.5.2 Entitlement Assignments

The Entitlement Assignments scheduled task is used for copying data about assigned entitlements into the ENT_ASSIGN table, in case when triggers fail to synchronization entitlement from UD table to ENT_ASSIGN. This task identifies the entitlement attribute from the child process form table, and then copies data about assigned entitlements from the child process form table into the ENT_ASSIGN table. A record created in the ENT_ASSIGN table corresponds to an entitlement assigned to a particular user on a particular target system.

You can use the RECORDS_TO_PROCESS_IN_BATCH attribute of this scheduled task to specify the number of records in each batch. The default batch size is 5000.

In addition, it creates INSERT, UPDATE, and DELETE triggers on the child process form tables from which it copies entitlement data.

10.4.6 Deleting Entitlements

Entitlements can get deleted in any one of the following ways:

- Deleting the Entitlement in the target, followed by synchronizing it via lookup reconciliation and further by the Entitlement List schedule job.
- Direct deletion of the Entitlement from Entitlement List via APIs.
- Deleting via corresponding application instance.

In all the ways of deleting, the Entitlement is marked as soft-deleted, that is, the "valid" flag on the Entitlement is updated to mark it as soft-deleted.

In all the cases of deleting, you need to perform the following post-processing.

- Unpublish the entitlement from the organization to which it is published
- Update the Modify_date on the Entitlement in Entitlement List to the current date
- Purge the instances of the Entitlement in the child table and Entitlement Assign
- Remove the Entitlements that are picked up by Catalog harvesting, that are marked as soft-deleted, and all request profiles.

Note:

- In-flight requests that have references to soft-deleted Entitlements will fail.
- Access Policies having deleted Entitlements should be manually updated to remove the same.

To perform post-processing of Entitlement soft-deletion in the provisioning component:

1. Run the `Entitlement Post Delete Processing Job` scheduled job.

   This task will take the following inputs:

   - Application Instance Name/ALL
   - Mode: Revoke/Delete

2. The task will perform the following functionality:

   - Revoke mode: The scheduled task will revoke the entitlement-grant for all the accounts in Oracle Identity Manager, which have that specific entitlement granted.
   - Delete mode: The schedules task will simply hard-delete the entitlements from Oracle Identity Manager database in the UD_CHILD table.
   - In both the above cases, the Entitlement grant entry is removed from ENT_ASSIGN.

   Note: The `Mode` flag must be set to `Delete`, and not `Revoke`, when you want to compensate for the post deletion of the entitlements. If you want that the entitlements being deleted from the backend through the Design Console should also be removed from the request details, and the Grand task and the Revoke task should not appear in the user's inbox, then you must run the `Entitlement Post Delete Processing Job` scheduled job with the `Mode` flag set to `Delete`.

3. Run the `Entitlement List` scheduled task. This is an existing schedule task that will go to all the resources that have an entitlement field, get the corresponding lookup definition and populate ENT_LIST with the values from the lookup definition, setting the correct SVR_KEY in the process.

10.4.7 Refreshing the Entitlement List Post Delete for New Entries

When an entry with the same encoded value is deleted and added consecutively in a lookup code, you need to perform the following steps to synchronize the data to the entitlement list:

1. Login to Oracle Identity System Administration.
2. Run the Entitlement List job to soft delete the existing entry.
3. Run the Entitlement Post Delete Processing Job scheduled job with Delete mode to clean up soft deleted items.
4. Run Entitlement List job again to add the new entry.

10.4.8 Disabling the Capture of Modifications to Assigned Entitlements

You can manually disable incremental synchronization of assigned entitlement data in the ENT_ASSIGN table. In other words, you can disable the capture of modifications to assigned entitlements. To achieve this, you create and run an SQL script to drop the following triggers created on the child process form tables:

Note: These triggers are created by the Entitlement Assignments scheduled task.

- The OIU_UDPATE trigger created on the OIU table
- The *TABLE_NAME_ENT_TRG* triggers created on the UD_ tables:

After you run the script, modifications to assigned entitlements are not copied into the staging table.

The following is a sample SQL script to drop the triggers on the child process form tables:

```
create or replace
TRIGGER UD_LDAP_GRP_ENT_TRG
AFTER INSERT
OR DELETE
OR UPDATE OF UD_LDAP_GRP_GROUP_NAME
ON UD_LDAP_GRP
FOR EACH ROW
BEGIN
CASE
WHEN INSERTING THEN
OIM_SP_MANAGEENTITLEMENT('UD_LDAP_GRP',:NEW.UD_LDAP_GRP_GROUP_NAME,NULL,
:NEW.UD_LDAP_GRP_KEY,:NEW.ORC_KEY,NULL,NULL,NULL,
NULL,NULL,'INSERT');
WHEN UPDATING THEN
IF :NEW.UD_LDAP_GRP_GROUP_NAME != :OLD.UD_LDAP_GRP_GROUP_NAME
THEN
OIM_SP_MANAGEENTITLEMENT('UD_LDAP_GRP',:NEW.UD_LDAP_GRP_GROUP_NAME,
:OLD.UD_LDAP_GRP_GROUP_NAME,:NEW.UD_LDAP_GRP_KEY,:NEW.ORC_KEY,NULL,
NULL,NULL,
NULL,NULL,'UPDATE');
END IF;
WHEN DELETING THEN
OIM_SP_MANAGEENTITLEMENT('UD_LDAP_GRP',:OLD.UD_LDAP_GRP_GROUP_NAME,
NULL,NULL,:OLD.ORC_KEY,NULL,NULL,NULL,
NULL,NULL,'DELETE');
END CASE;
END;
```

10.4.9 Entitlement-Related Reports

The following predefined reports provide data about assigned entitlements:

Note:

You must be a member of the ADMINISTRATORS group to be able to view these reports.

Duplicate assignments of the same entitlement to a particular user are suppressed in the reports because they are not copied to the ENT_ tables. For example, if user John Doe has been assigned the Sales Superintendent role twice on a target system, then the reports show only one instance of this entitlement.

- Entitlement Access List
- Entitlement Access List History
- User Resource Entitlement
- User Resource Entitlement History

10.4.9.1 Entitlement Access List

The Entitlement Access List report lists users who are currently assigned the entitlements that you specify while generating the report. The report provides basic information about the entitlements and the list of users to whom the entitlements are assigned.

10.4.9.2 Entitlement Access List History

The Entitlement Access List History report lists users who had been assigned the entitlements that you specify while generating the report. The report provides basic information about the entitlements and the list of users to whom the entitlements were assigned.

10.4.9.3 User Resource Entitlement

The User Resource Entitlement report lists the current entitlements of users whom you specify while generating the report. The report displays basic user information and entitlement details.

10.4.9.4 User Resource Entitlement History

The User Resource Entitlement History report lists details of past entitlements assigned to users whom you specify while generating the report. The report displays basic user information and entitlement details.

10.5 Managing Disconnected Resources

Disconnected resources are targets for which there is no connector. Therefore, the provisioning fulfillment for disconnected resources is not automated, but manual. In earlier releases of Oracle Identity Manager, disconnected provisioning is not supported as a first class use case, it is supported by using manual tasks in the provisioning process. This approach has a number of limitations, which are taken care in Disconnected Resources model. In Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0), disconnected resources are an enhanced configuration for manual provisioning that leverage SOA integration to provide higher flexibility and configurability of the manual provisioning workflow.

Some examples of disconnected resources include a Badge, Laptop, Pager, or any such item wherein the fulfillment is manual.

This section enlists the following topics:

- Disconnected Resources Architecture
- Managing Disconnected Application Instance
- Provisioning Operations on a Disconnected Application Instance
- Managing Entitlement for Disconnected Resource
- Status Changes in Manual Process Task Action
- Customizing Provisioning SOA Composite
- Troubleshooting Disconnected Resources

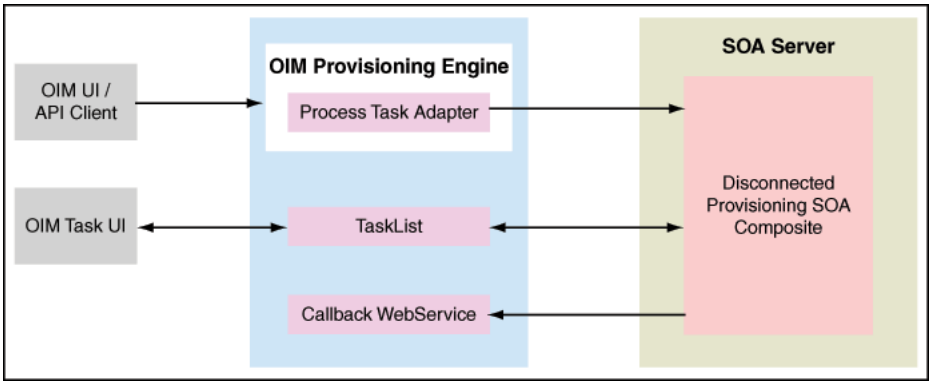
10.5.1 Disconnected Resources Architecture

The Disconnected Resource feature makes use of the existing Oracle Identity Manager provisioning engine artifacts such as the Provisioning Process, Process Task, Adapters and so on while providing BPEL Integration in a seamless and configurable manner.

When a Disconnected Application Instance is created from the UI, it automatically seeds a number of backend configuration artifacts, including a resource object (of type Disconnected), a provisioning process with tasks for the basic provisioning operations, an IT resource, and a process form with the minimal fields (which can be further customized).

Figure 10–3 illustrates the provisioning process architecture for disconnected resources.

Figure 10–3 Disconnected Resource Architecture



When a disconnected application instance is provisioned to a user (via request or otherwise), the specific workflow in the provisioning process is triggered. This fires the corresponding process task and executes the manual provisioning adapter that invokes the out of the box disconnected provisioning SOA composite. A SOA manual task is assigned to System Administrator by default. When the assignee acts on the manual task, the provisioningcallback webservice is invoked with the assignee specified response and it then completes or aborts the provisioning operation and updates the account appropriately.

Table 10–4 displays the attributes for manual provisioning SOA composite payload that is available in the composite.

Table 10–4 Manual Provisioning SOA Composite Payload Attributes

Attribute	Description
Account ID	Account ID (oiu_key) for the account under consideration
AppInstance Name	Disconnected Application Instance Display Name
Resource Object Name	Disconnected Resource Object Name
ITResource Name	Disconnected ITResource Name
Beneficiary Login	Login of the account beneficiary
Entity Key	Application Instance Key in case of Provision, Revoke, Disable, and Enable account operations.
Entity Type	Type is set to ApplicationInstance, in case of Provision, Revoke, Disable, and Enable account operations.
Beneficiary First Name	First name of the account beneficiary
Beneficiary Last Name	Last name of the account beneficiary
Descriptive Field	Account descriptive field for the account under consideration
URL	Oracle Identity Manager callback URL for the webservice.
Request Key	Request Key if operation is through request.
Requester Login	Login of the requester if operation is through request.

10.5.2 Managing Disconnected Application Instance

Managing disconnected application instance includes the following tasks:

- Creating a Disconnected Application Instance
- Creating a Disconnected Application Instance for an Existing Disconnected Resource

10.5.2.1 Creating a Disconnected Application Instance

Note: You must create a new sandbox before creating the application instance. You must publish the sandbox after creating the application instance. See "Managing Sandboxes" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about creating and publishing a sandbox.

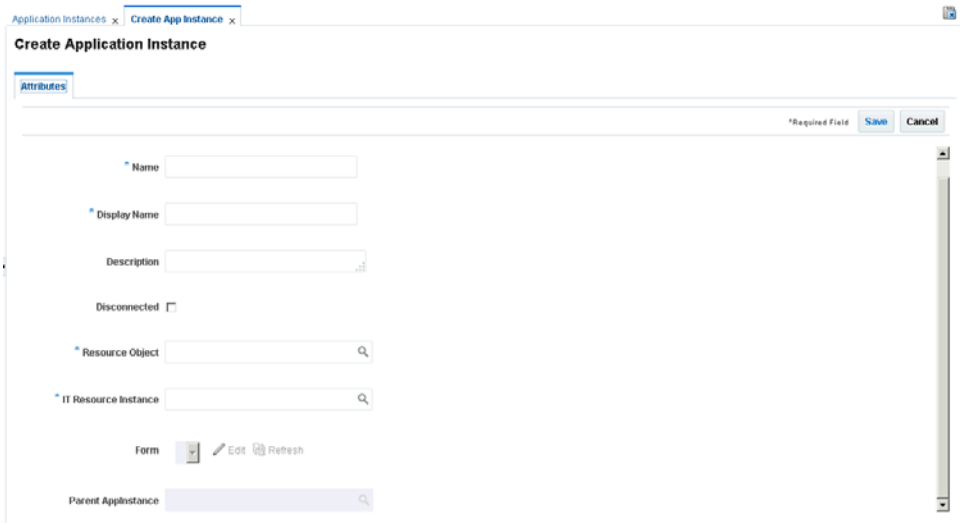
To create disconnected application instance:

1. Log in to Oracle Identity System Administration.
2. Create and activate a sandbox.
3. In the left pane, under Configuration, click **Application Instances**. The Application Instances page is displayed.
4. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Application Instance page is displayed.
5. In the respective attribute fields, enter the values as shown in the following table:

Attribute	Value
Name	Enter the name of the application instance. This is a required field.
Display Name	Enter the display name of the application instance. This is a required field.
Description	Specify a description of the application instance.
Disconnected	Select the checkbox. This is the flag to indicate whether the application instance is not connected. Note: This is a UI only flag and is not persisted in the backend. Checking this flag will disable Resource Object and ITResource Instance fields, as these are automatically created in the back end.

Figure 10–4 shows the attributes for Create Application Instance.

Figure 10–4 Create Application Instance Attributes



6. Click **Save**, and then click **OK** on the information dialog box. The application instance is created, and the details of the application instance is displayed.
7. Publish the sandbox.
8. The UI form for the disconnected resource is automatically created and set, click **Apply**.
9. In addition to the application instance, in the back end, the following provisioning artifacts are automatically created:
   - Resource object of type Disconnected
   - ITresource type definition with the following parameters:
     - Configuration Lookup
     - Connector Server Name
     - Identity Gateway Name

Note: IT resource type definition parameters are for future use and the values for the same need not be set.

- IT resource of type definition
- Parent process form with the following fields:
  - Account ID
  - Password
  - Account login
  - IT resource
- Process definition with workflows for the following operations:
  - Provision Account
  - Enable Account
  - Disable Account
  - Revoke Account
  - Modify Account Attributes
- Adapters
  - Manual Provisioning
  - Manual Entitlement Provisioning

10. From the System Administration UI, search for schedule job called "Catalog Synchronization Job" and execute it.

10.5.2.2 Creating a Disconnected Application Instance for an Existing Disconnected Resource

To create a disconnected application instance for an existing disconnected resource, see "Creating Application Instances" on page 10-4.

> **Note:** You must not select the **Disconnected** option, as this will create artifacts including the resource object and IT resource in the backend.

10.5.3 Provisioning Operations on a Disconnected Application Instance

When provisioning process is triggered for Enable, Disable, Revoke, or Provision operations, the corresponding process task is inserted which runs the Manual Provisioning adapter. This adapter invokes the out of the box provisioning SOA composite. A SOA Human Task is assigned to the System Administrator by default.

From the Inbox in Oracle Identity Self Service, the System Administrator can:

- Check the task details
- Check the account details
- Change process form data in Oracle Identity Manager by changing data and clicking the Fulfill button
- Perform the operation manually in the target
- Act on the pending task by clicking Complete or Reject.

When the assignee acts on the pending manual tasks, the provisioning callback webservice is invoked which continues with the Oracle Identity Manager operation and updates the account appropriately. See "Status Changes in Manual Process Task

Action" on page 10-32 for details on changes to account status based on assignee action.

Oracle Identity Manager does not support the following provisioning operations on a disconnected application instance:

- Password operations
- Provisioning process customization operations

10.5.3.1 Process Form Updates

When a process form field of a disconnected resource is updated, the "<FORM_NAME> Updated" process task is inserted into the provisioning process. This would generate a manual SOA human task, so that the assignee can manually update the changes in the corresponding target.

> **Note:** The "<FORM_NAME> Updated" task is inserted irrespective of whether updates are to a single process form field or multiple process form field. This behavior is different from that of a connected resource. In addition, note that the individual process form field update tasks need not be configured for a disconnected resource.

10.5.4 Managing Entitlement for Disconnected Resource

Managing entitlement for disconnected resource includes the following:

- "Configuring Entitlement Grant" on page 10-30

10.5.4.1 Configuring Entitlement Grant

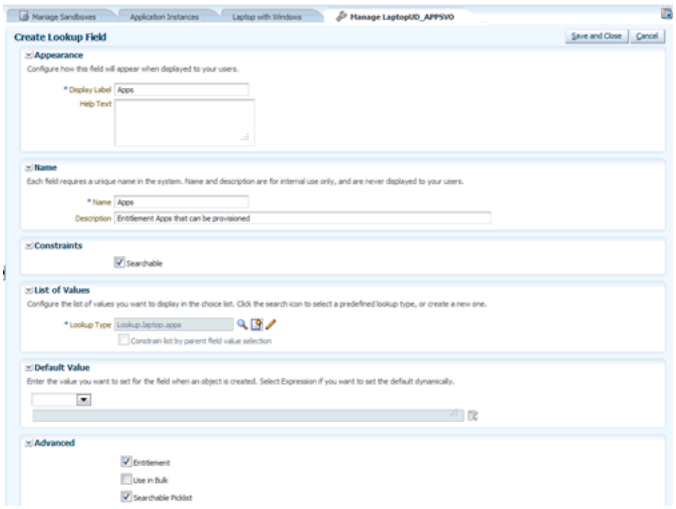
Configuring entitlement grant for disconnected resource involves creating a child form and configuring the lookup definition for entitlements, which is as follows:

> **Note:** Before creating child forms, create and activate a sandbox.

1. Go to Oracle Identity System Administration. Under Configuration, click **Form Designer** and perform the following steps:
   a. Click on the Resource Type and search for the Disconnected Resource.
   b. From the search result, click on the disconnected application instance form name.
2. Go to Child Objects tab and click **Add** to add a child form.
3. In the Name field, provide a name to the child table and click **OK**.
4. Click the name link to open it for editing.
5. Click **Create**. In the Select Field Type dialog box, select **Lookup**, and click **OK**.
6. Provide the following values for the entitlement field:
   a. In the Display Label field, enter a display name.
   b. In the Name field, enter a name for the lookup.
7. Select the following check boxes:
   - Searchable

- Entitlement
- Searchable Picklist

Note: It is mandatory that you must select Searchable, Entitlement, and Searchable Picklist check boxes to create an entitlement field on the child form.



8. Create a new custom field of Lookup Type and click **OK**.
9. In the List of Values section, click the create a new lookup type icon and provide values for Meaning (for example, Lookup.Laptop.apps), Code (for example, Lookup.Laptop.apps) and description as follows:
   a. Click new to add entitlement values to add Lookup Codes. The value in the Code and Meaning columns should have the following format:

Code	Meaning
<ENTITTLEMENT_NAME>	<ENTITLEMENT_DESCRIPTION>

   b. Click **Save**. The Create Lookup Type dialog box closes.
   c. Click **Save and Close**.
10. Click **Back to Parent Object** to return to the parent form.
11. Click **Regenerate View** to regenerate UI artifacts and dataset, and confirm by clicking **OK**.

    See "Modifying Forms By Using the Form Designer" on page 6-3 for information about the options available in the Regenerate View popup window.
12. Publish the sandbox.
13. Go back to Oracle Identity System Administration, System Management, Scheduler.
14. Search for a scheduled job called Entitlement List and execute it.

15. After the scheduled job execution completes, search for another schedule job called Catalog Synchronization Job and execute it.

Note: Customization of the provisioning process is not supported, but you can customize the Disconnected Provisioning Composite.

10.5.5 Status Changes in Manual Process Task Action

Table 10–5 provides details about status changes based on manual task action:

Table 10–5 Manual Process Task Action Statuses

Provisioning Operation	Manual Task Action	Provisioning Action
Provision	Complete	Account status is set to Provisioned.
Provision	Reject	Account status is not updated.
Disable	Complete	Account status is set to Disabled.
Disable	Reject	Account status is not updated.
Enable	Complete	Account status is set to Enabled.
Enable	Reject	Account status is not updated.
Revoke	Complete	Account status is set to Revoked.
Revoke	Reject	Account status is not updated.
Update	Complete	No Operation
Update	Reject	No Operation
Grant Entitlement	Complete	Completes the child table insert trigger process task and sets entitlement status to Provisioned.
Grant Entitlement	Reject	Cancels the child table insert trigger process task, which deletes the child table entry.
Revoke Entitlement	Complete	Deletes the child table entry from Oracle Identity Manager.
Revoke Entitlement	Reject	No Operation

10.5.6 Customizing Provisioning SOA Composite

Provisioning SOA composite includes the following customizations:

- Customizing Human Task Assignment via SOA Composer
- Customizing by Modifying the Out of the Box Composite

10.5.6.1 Customizing Human Task Assignment via SOA Composer

The manual disconnected provisioning SOA composite, has a default rule, ManualProvisioningRule, which assigns the human task to the System Administrator.

A custom rule with higher priority, based on the payload, for example Application Instance Name, can be created from the SOA Composer UI, based on which the manual task assignment can be customized.

To add a custom rule:

1. Access Oracle SOA Composer by navigating to the following URL:

http://*SOA_HOST*:*SOA_PORT*/soa/composer

2. Log in to the SOA Composer UI and click **Open Task** and select `DisconnectedProvisioning_rev1.0 composite.`
3. From the ManualProvisioningTaskRules.rules tab, click **Edit** to add a custom rule.
4. Add Rule by providing the rule name and the conditional assignment rule.
5. Using the Up arrow, move the custom rule above the ManualProvisioningRule.
6. Save and commit changes. The manual provisioning rule is added.

> **See Also:** SOA Composer documentation for more information about creating rules

10.5.6.2 Customizing by Modifying the Out of the Box Composite

To modify the out of the box Disconnected Provisioning composite:

1. Copy the composite from *OIM_HOME*/workflows/composites/DisconnectedProvisioning.zip to a local JDeveloper working location. Unzip it in the same directory to create the DisconnectedProvisioning directory.
2. Open the composite in JDeveloper in Default Role.

> **Note:** You must install the version of JDeveloper that is compatible with the Oracle Identity Manager deployment. In addition, install any patches for JDeveloper so that JDeveloper works correctly with the SOA composites.

3. As part of customization do not alter the following:
   - Payload attributes defined in DisconnectedProvisioning\xsd\ManualProvisioningTaskPayload.xsd
   - ProvisioningCallbackService partnerlink and mappings
4. Double-click **composite.xml** to open the composite and modify as per your requirements.
5. Deploy the SOA composite from Jdeveloper to Oracle SOA server. Make sure that you do not update the Revision ID and select the Overwrite any existing composites with the same revision ID option.

10.5.7 Troubleshooting Disconnected Resources

Table 10–6 displays the common problems that you may encounter while performing provisioning and other tasks for disconnected resources.

Table 10–6 Troubleshooting Disconnected Resources

Problem	Solution
Upon provisioning disconnected application instance, manual task is not assigned to assignee.	Perform the following steps: 1. Make sure that the SOA server is running. 2. Check Open tasks page for rejected process tasks, and check the error information in the task, if it exists. 3. Check Oracle Identity Manager logs to check if adapter is running.
Upon manual task completion, account status is not modified.	Perform the following steps: 1. Make sure that the provisioning callback webservice, Provcallback is deployed. 2. Test the Webservice from the application server console.

11

Managing Connector Lifecycle

Oracle Identity Manager offers various solutions for integration with different kinds of IT-based resources in an organization. Oracle Identity Manager connectors are the recommended solution for integration between Oracle Identity Manager and resources that store and use user data. A connector enables exchange of user data between Oracle Identity Manager and a specific resource or target system.

Oracle Identity Manager server uses connectors to perform operations on target systems. Oracle provides connectors for common enterprise resources. You can develop custom connectors for your own resources.

A connector consists of the following artifacts:

- Binaries (JAR and DLL files) that contain the connector code
- XML file(s) consisting of data of Objects defined in Oracle Identity Manager, such as an IT resource, resource object, provisioning process and process tasks, process form and child forms, adapters and adapter tasks, lookup definitions, reconciliation rules, and scheduled tasks
- Integration libraries that enable adapters to perform actions on the target system

  For some target systems, third-party integration libraries might be required to enable communication or specific functionality with the target systems.

  See Also: *Oracle Identity Manager Connector Concepts* for detailed conceptual information about connectors and connector objects

This chapter provides information about Connector Lifecycle Management (LCM) features. It is divided into the following sections:

- Troubleshooting Connector Management Issues

11.1 Lifecycle of a Connector

The following are stages in the lifecycle of a connector:

Deployment

A connector can be installed by clicking the **Manage Connector** menu on the Advanced Administration section of the Oracle Identity System Administration.

To complete the deployment procedure, you might also need to copy connector files and external code files to destination directories on Oracle Identity Manager and target system host computers. Some connectors require a Remote Manager, which is usually installed on the target system host computer. Some other connectors, specifically the identity connectors, require the local and remote connector server.

Oracle Identity Manager 11.1.1.5.*x* provide Connector LCM as a new feature to manage connectors and uses Connector Installer (CI) for installing connector.

Installing a connector using Connector Installer is not the same as doing it using Deployment Manager. Although the Deployment Manager offers an alternative approach to import definitions of the objects that constitute a connector, the connector imported using Connector LCM can be managed better as Connector LCM offers a more broader and richer feature than Deployment Manager. Therefore, the Install Connectors feature is the recommended approach for Oracle Identity Manager 11g based connector installation and/or management.

See Also:

- Oracle Identity Manager Connector documentation for information about copying connector files and external code files to destination directories on Oracle Identity Manager and target system host computers. Connector documentation is available on the Oracle Web site at the following URL:

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

- "Understanding Identity Connector Framework" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about the Identity Connector Framework and how to use it to create an identity connector.

Customization

After deployment, you might customize a connector to meet business requirements that are not addressed by the default configuration of the connector. For example, you might add new attributes for reconciliation and provisioning with the target system. An enhancement of this type requires changes to be made in multiple connector objects, such as Resource Object, Process Definition, and Process Form. See Connector Documentation for detailed information about changes required in connector objects.

Cloning

You might have more than one installation of a target system. If you have a target system with multiple instances, and data is either same or shared or replicated, such as in Microsoft Exchange or Active Directory connectors, then you do not need to clone the connector. You need to create multiple IT resources for the instances. The target works as a single resource object.

If you have a target system with different installations or schema or data, such as a LDAP server for internal users and another LDAP server for external, contractors, and consumers, then you need to clone the connector. The connectors will work as two separate targets.

There might be a scenario where the connector attributes are different. Then instead of creating a new connector, the existing connector can be cloned by using the XML of the original connector. The **Clone Connectors feature** of the Advanced Administration enables you to automatically generate copies of a set of connector objects.

Upgrade

To make use of new features introduced in later releases of a connector, you might upgrade a connector by applying patch sets released by Oracle. Typically, upgrading to a new release of a connector involves processes that range from simple changes (such as a JAR file upgrade) to changes that affect most of the adapter tasks that were shipped as part of the connector. You can use the **Upgrade Connectors feature** to upgrade a connector.

Note: Upgrading connectors preserve the existing customizations in a connector.

Uninstalling

Note: Uninstalling a connector is performed in the development environment and not in production environment.

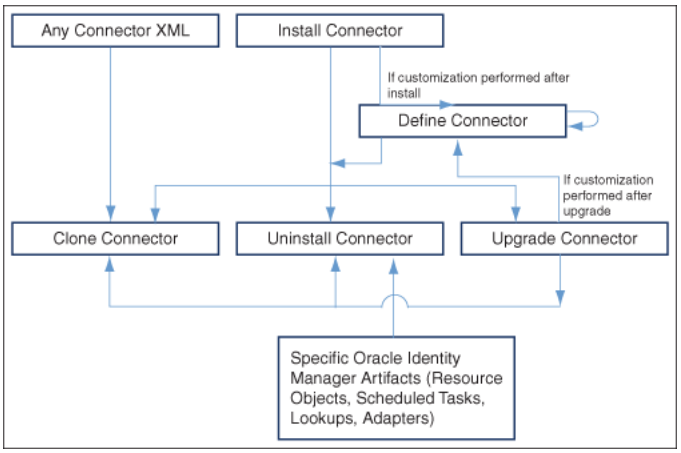
If you stop using a connector, then this action is also provided to additional environments, such as System Integration Testing, User Acceptance Testing, and Staging, where that connector is also stopped.

The need to keep a clean development environment that does not have any unnecessary Oracle Identity Manager objects, you would like to uninstall a particular connector version that you no longer need to use. The **Uninstall Connectors utility** enables you to uninstall connectors as well as individual connector objects.

Note: You must have the System Administrator role to perform connector lifecycle management tasks, such as installing connectors including importing connector XML files by using the Deployment Manager, and cloning, defining, upgrading, and uninstalling connectors.

Figure 11–1 depicts the connector lifecycle:

Figure 11–1 Connector Lifecycle



11.2 Connector Lifecycle and Change Management Terminology

The following terms have been introduced in this chapter:

- **Oracle-released connector** refers to a connector released by Oracle.
- **Custom release** or **custom connector** refers to connectors that you develop as well as Oracle-released connectors that you customize or reconfigure in any way.
- **Source release** or **source connector** refers to the existing release of the connector that you want to upgrade to a different (that is, new) release. For example, if you want to upgrade the SAP User Management connector from release 9.1.2 to release 9.1.2.1, then release 9.1.2 is the source release.
- **Target release** or **target connector** is the release to which you want to upgrade the source release. In the preceding example, SAP User Management release 9.1.2.1 is the target release.

> **Note:** Some of the preceding terms can be combined to provide a shortened description of the type of connector that is under discussion. For example, a **custom source release** is a connector that you had created, customized, or reconfigured and now want to upgrade to a target release.

- A **configuration XML file** contains information that is used during connector installation by the Install Connectors feature. For a connector released by Oracle, the configuration XML file is included in the deployment package. For a custom-developed connector, you might want to develop the individual connector objects on the staging (test) server and then deploy the connector on the production server. In this case, you can create a configuration XML file for the connector if you want to install the connector on the production server by using the Install Connectors feature.

  > **See Also:** "Installing Connectors" on page 11-6 for information about the Install Connectors feature.

- A **connector XML file** contains definitions of the individual objects that constitute a connector. When the XML file is imported into Oracle Identity Manager through the Deployment Manager, these objects definitions are used to create the connector

objects in the Oracle Identity Manager database. The manner in which the XML file is imported into Oracle Identity Manager depends on the type of connector:

- For an Oracle-released connector that is compatible with the Install Connectors feature, the connector XML file is automatically imported when you use the Install Connectors feature. This feature implicitly calls the Deployment Manager to import the connector XML file.
- For an Oracle-released connector that is not compatible with the Install Connectors feature, you use the Deployment Manager to import the XML file.
- For a custom connector, you can use the Deployment Manager to first export definitions of objects that you had created on the staging server. The output of this process is the connector XML file. You can then import the file into the production server. Alternatively, if you create a complete deployment package (including the configuration XML file) for the connector, then you can use the Install Connectors feature to install the connector. This feature implicitly calls the Deployment Manager to import the file.

See Also: "Exporting Connector Object Definitions in Connector XML Format" on page 11-33 for information about exporting connector object definitions by using the Deployment Manager

11.3 Viewing Connector Details

To view the details of a connector:

Note: In this release of Oracle Identity Manager, the connector lifecycle management functionality have been introduced such as defining, cloning, upgrading, and uninstalling connectors. For all these features, complete connector DM-XML is required in the database, and this is the source for all the connector lifecycle management activities.

When Oracle Identity Manager is upgraded from earlier releases, such as Release 9.1.*x* or 11*g* Release 1 (11.1.1.5), to 11*g* Release 2 (11.1.2.3.0), you must define the connector so that all the lifecycle management operations on the connector are possible to perform. Without defining the connector, it is not possible to search for the installed connector, upgrade the installed connector, clone the connector, and uninstall the connector. See "Defining Connectors" on page 11-12 for information about defining connectors.

---

1. Login to Oracle Identity System Administration.
2. In the left pane, under Provisioning Configuration, click **Manage Connector**.
3. In the **Connector Name** field, enter the name of the connector.
4. Click **Search**. The search result shows the details of the connector.

   If you do not know the full name of the connector, then you can perform a wildcard search for a connector. For example, if you want to display details of the Microsoft Active Directory connector installed in your operating environment, then you can use "*Direct*" as the search string.

   If you want to display details of all installed connectors, then you can leave the Connector Name field blank and click **Search**.

The search results table displays the connector name, release number, status, and the date and time at which the connector was installed. The remaining columns of the table provide icons that you can use to begin any of the lifecycle management operations on a connector.

11.4 Installing Connectors

The following sections describe this feature and the procedure to use it:

Note: To determine whether you can install an Oracle-released connector by using the Install Connectors feature, see the connector guide.

- Overview of the Connector Deployment Process
- Creating the User Account for Installing Connectors
- Installing a Connector

11.4.1 Overview of the Connector Deployment Process

To install a connector, you perform some or all of the following tasks:

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

Of these tasks, the Install Connectors feature automatically performs the following:

Note: You manually perform the remaining tasks. Connector documentation provides instructions.

- 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)

At the end of a successful installation, an entry is created in a table in the Oracle Identity Manager database that stores data about installed connectors. "Defining Connectors" on page 11-12 describes the data that is stored in the database.

11.4.2 Creating the User Account for Installing Connectors

Users belonging to the SYSTEM ADMINISTRATORS role of Oracle Identity Manager can install connectors.

11.4.3 Installing a Connector

Note:

- Re-installing a connector is not supported. You cannot install a connector version that had already been installed in Oracle Identity Manager. However, if the installation process is not successful, Oracle Identity Manager allows you to reinstall the connector.
- Before installing a connector, create a backup of your environment. This is because, if the installation fails, then the connector cannot be uninstalled. There is no solution for reverting the environment to the previous state or finalizing the connector install.

Before you install a connector, copy the installation files of the connectors that you want to install into the default connector installation directory, which is:

```
OIM_HOME/server/ConnectorDefaultDirectory
```

To install a connector:

1. Log in to Oracle Identity System Administration by using the user account described in "Creating the User Account for Installing Connectors" on page 11-7.
2. In the left pane, under Provisioning Configuration, click **Manage Connector**.
3. Click **Install** in the top-right corner of the page.
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/server/ConnectorDefaultDirectory
   ```

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

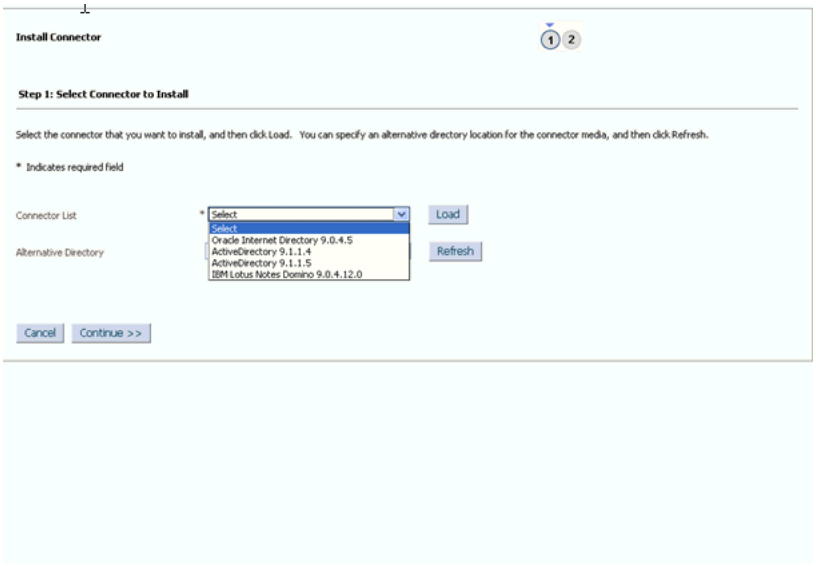
   a. In the **Alternative Directory** field, enter the full path and name of that directory.

   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.

   Figure 11–2 shows the Select Connector to Install page of the Install Connector wizard:

Figure 11–2 The Select Connector to Install Page



5. Click **Load**.

   The following information is displayed:

   - Connector installation history

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

   - Connector dependency details

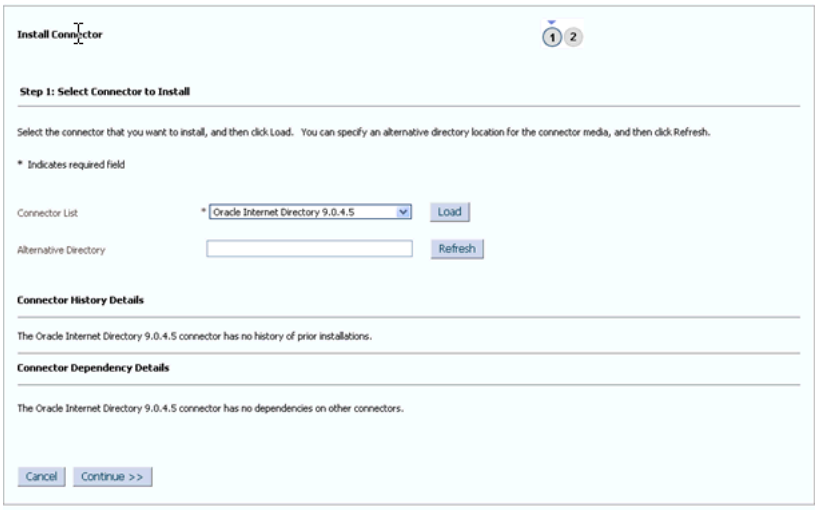
     There are some connectors that require the installation of some other connectors before you can start using them. For example, before you use the Novell GroupWise connector, you must install the Novell eDirectory connector. Novell eDirectory is called the **dependency connector** for Novell GroupWise.

     The connector dependency details include the list of connectors that must be installed before you can install and use 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. However, after showing the dependency information, the Install Connector wizard allows you to install the connector.

     You must ensure that the correct versions of dependency connectors are installed after you complete the current installation.

     Figure 11–3 shows the page with connector history details and connector dependency details:

Figure 11–3 Connector History and Dependency



6. To start the installation process, click **Continue**.

Note: The Install progress screens might flash and show blank page. This does not have any impact on functionality and can be ignored.

The following tasks are performed in sequence:

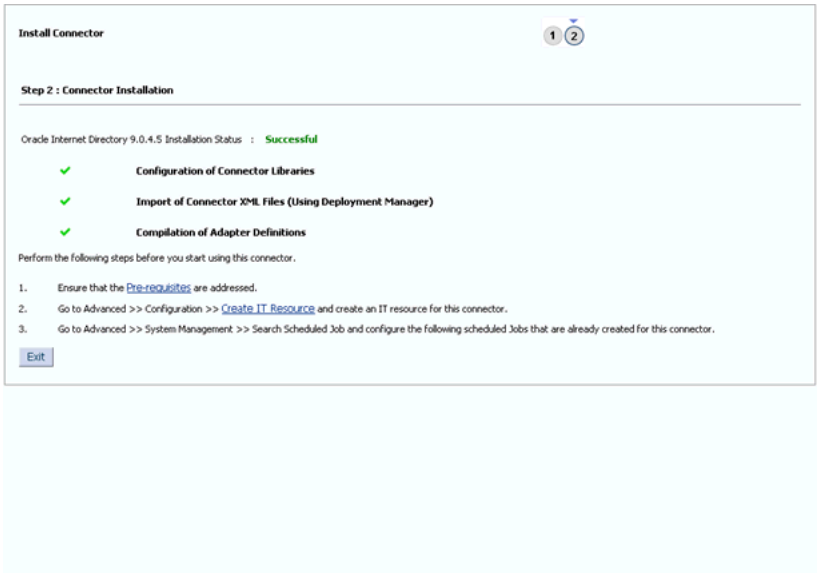
a. Configuration of connector libraries

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

c. Compilation of adapters

Figure 11–4 shows the Connector Installation page of the Install Connector wizard:

Figure 11–4 The Connector Installation Page



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:

- Fix the cause of the error, and then retry installation by clicking **Retry.**
- Cancel the installation and begin again from step 1 of the installation procedure.

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 *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. 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: There are no prerequisites for some connectors.

   b. Creating an IT resource for the connector

      Most of the connectors are shipped with a default IT resource. You can use either the default IT resource or create a new one. To create a new IT resource, go to System Administration, under Configuration, click **IT Resource.** The Manage IT Resource page opens. On this page, click **Create IT Resource**.

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

      To configure scheduled task, go to System Administration, under System Management, click **Scheduler** and search for required scheduled job.

11.4.4 Post Installation Steps

To perform post installation configuration:

1. Create or update IT resource with appropriate values using steps defined in 7-b.

2. Creating a Sandbox. To do so:

   See Also: "Managing Sandboxes" section in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for complete information on Sandboxes

   a. Navigate to System Administration and on the top right hand corner, click **Sandboxes**.

   b. In the Manage Sandboxes tab, click **Create Sandbox.**

   c. In the Create Sandbox dialog box, enter a sandbox name and description, click **Save and Close.** Click **Ok** in the confirmation dialog box.

3. Creating a new UI form. To do so:

   See Also: see Chapter 6, "Managing Forms." for complete information about forms.

   a. In the System Administration page, under Provisioning Configuration, click **Form Designer**.

   b. Under Search Results, click **Create**.

   c. Select the resource type for which form needs to be created.

   d. Enter a form name and click **Create.**

4. Creating an Application Instance. To do so:

   See Also: see Chapter 10, "Managing Application Instances." for complete information about Application Instances.

   a. In the System Administration page, under Provisioning Configuration, click **Application Instances**.

   b. Under Search Results, click **Create**.

   c. Enter appropriate values for fields displayed on the Attributes form and click **Save.**

   d. In the Form dropdown, select the newly created form and click **Apply.**

   e. Publish the application instance. See Section 10.2.3.2, "Managing Organizations Associated With Application Instances" for more information about publishing an application instance for a particular organization.

5. Export the sandbox and publish it.

   It is recommended that you export the sandbox to store all the changes made in your sandbox.

   For information about exporting and publishing sandboxes, see "Managing Sandboxes" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

6. Entitlement Harvesting and Catalog Sync:

   a. In the Identity System Administration, under System Configuration, click **Scheduler**.

   b. Run connector lookup reconciliation scheduled jobs.

   c. Run Entitlement List scheduled job.

d. Run Catalog Synchronization Job scheduled job.

11.5 Defining Connectors

Connector LCM operations such as Upgrade, Clone, and Uninstall needs a source for each connector where all the connector objects reside. The Connector Install stores the Deployment Manager (DM) XML in Oracle Identity Manager database.

Typically, you will install the shipped connector and then perform one or both of the following operations:

- Customize the connector by, for example, add/ modify existing object definitions, add additional adapters
- (Re) Configure the connector by, for example, changing attribute names and key fields

The DM XML in Oracle Identity Manager database, which is the reference for all Connector LCM operations need to be updated for customization changes. Oracle Identity Manager provides **Define** feature to update the DM XML stored in Oracle Identity Manager database with customization changes. Define feature is similar to Export where user need to add all the connector objects related to a specific connector. The end result of defining a connector is an XML file, which is updated in Oracle Identity Manager database.

At this point, the customized or re-configured connector is not the same as the Oracle-released connector. The connector XML file for the Oracle-released connector might not be valid for the customized or re-configured connector.

In the Advanced Administration page of the Oracle Identity System Administration, you can **define** a customized or re-configured connector. Defining a connector is equivalent to registering the connector with Oracle Identity Manager.

Note: You must add only those Oracle Identity Manager artifacts that are specific to the connector and do not add default objects or any other connector objects that are shared across connectors. The defined XML is the source for life cycle operations such as upgrade, clone, and uninstall. If an object is used in define and is shared across connectors or a default Oracle Identity Manager object, then there is un-intended behavior. For example, a Lookup Definition which is there by default in Oracle Identity Manager is added as a part of define, then clone operation will create another copy of the object, which is not required. The uninstall will delete this default object from Oracle Identity Manager as it is defined specific to a connector. Such incorrect definition will have impact on Oracle Identity Manager functionality. Therefore, you must be careful while adding an object while defining a connector.

When you define a connector, a record representing the connector is created in the Oracle Identity Manager database. If this record already exists, then it updates:

- The name of the connector. For example, `Microsoft Active Directory`.
- The release number of the connector. For example, `9.1.1`.
- The connector XML definitions.

Note:

- You can define the connector XML definitions in the form of an XML file. See the "Exporting Connector Object Definitions in Connector XML Format" section of the connector guide for more information. You can then use this connector XML file to build the installation package for installing the connector on a different Oracle Identity Manager installation.
- Oracle recommends defining a connector immediately after customizing the connector or updating the DM XML file with the customization changes.

A connector is automatically defined when you install it using the Install Connectors feature or when you upgrade it using the Upgrade Connectors feature. Therefore, if you install a connector and want to clone it without customizing the connector, then there is no need to define the connector.

You must manually define a connector, otherwise newer version (which basically pertains to entry in CIH table) of connector may not be reflected even though import of new XML was successfully completed. Perform this procedure only if:

- You import the connector by using the Deployment Manager.
- You customize or reconfigure the connector.

Note: You can continue to use a connector without defining it after you customize or reconfigure a connector or after you upgrade Oracle Identity Manager. However, if you want to upgrade, clone, or uninstall the connector, then you must first define it.

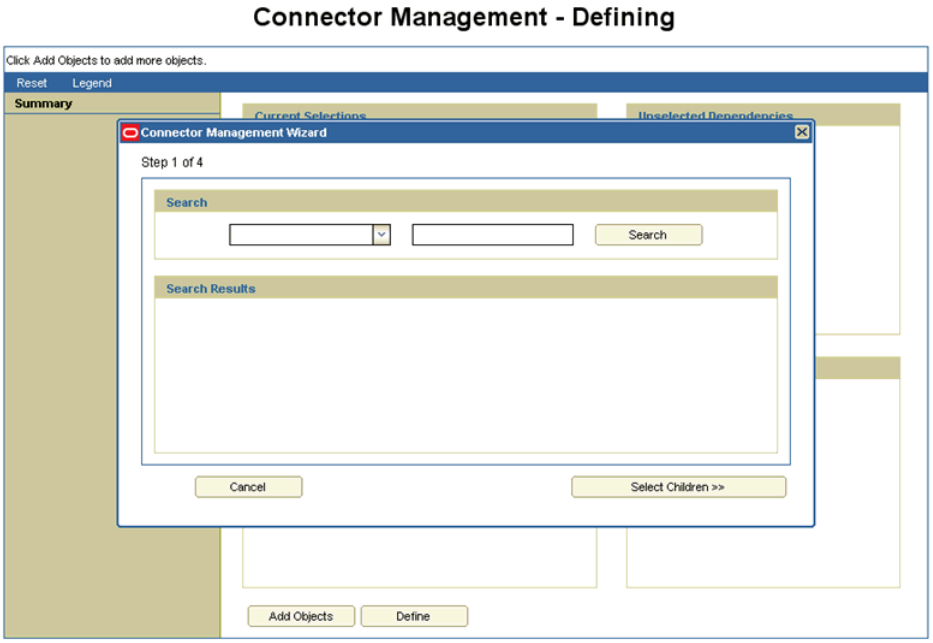
- You upgrade Oracle Identity Manager.
- It is a custom connector that you develop.

To define a connector:

Note: To determine whether you can define a particular release of a connector by using the Oracle Identity System Administration, see the documentation for that release of the connector.

1. Log in to Oracle Identity System Administration.
2. In the left pane, under Provisioning Configuration, click **Manage Connector**.
3. On the Connector Management window, click **Define**. The Connector Management Wizard is displayed, as shown in Figure 11–5:

Figure 11–5 Connector Management Wizard for Defining Connectors



4. On the first page of the wizard, select either **Resource** or **Process** from the Search list. In the adjoining field, you can enter a search string and the asterisk (*) as a wildcard character to refine your search for resource objects or process definitions belonging to the connector. Then, click **Search**.

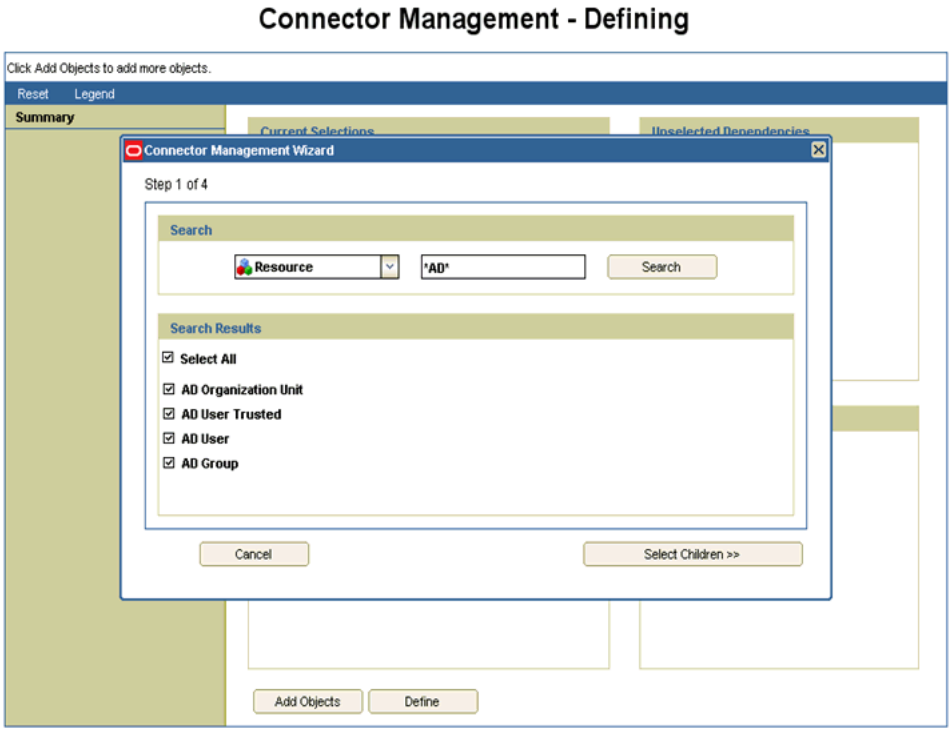
   Most of the objects that constitute a connector are linked to the resource objects and process definition of the connector. By selecting the resource objects or process definition, you automatically select the objects linked with them. Some of the connector objects, for example, scheduled task, do not have dependency with the resource object. Ensure that you search all the attributes and add them while defining.

   When you click Search, the list of resource objects or process definitions that meet the specified search criteria are displayed.

5. Select the check boxes for the resource objects or process definitions that are part of the connector.

   Figure 11–6 shows step 1 of the Connector Management Wizard with search results for connector objects:

Figure 11–6 Step 1 of the Connector Management Wizard

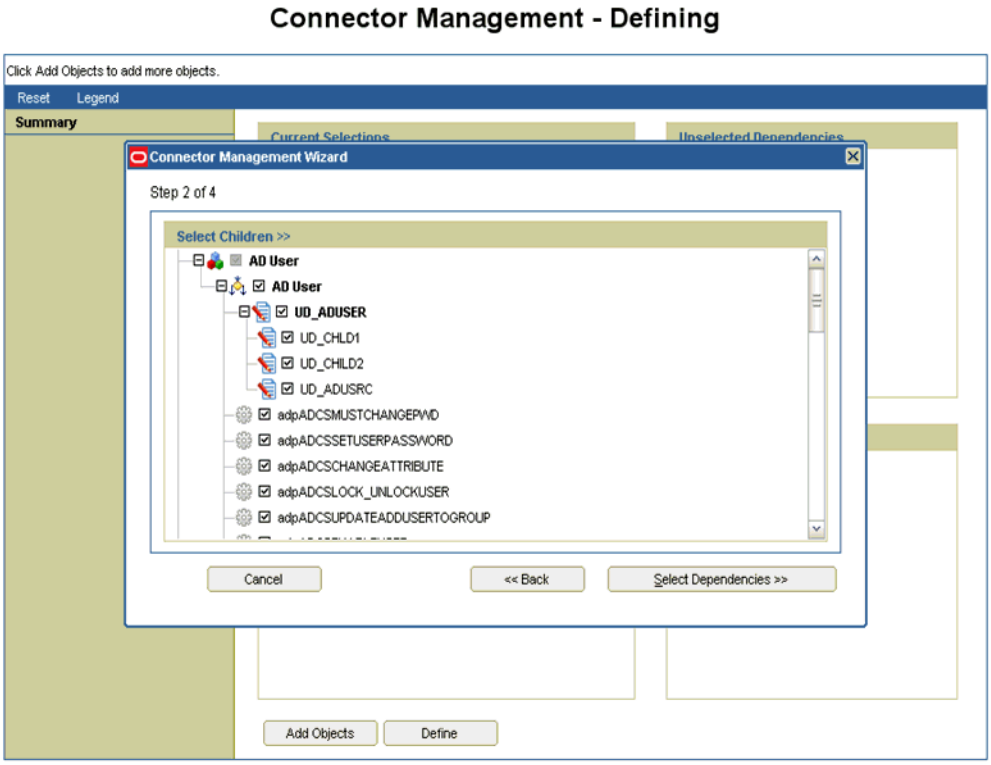


6. Click **Select Children**.
7. From the list of connector objects displayed, ensure that all the objects belonging to the connector are selected. Then, click **Select Dependencies**.

Note: For an Oracle-released connector, the adapters that are part of the connector are listed in the connector guide. Select the check boxes for those adapters.

Figure 11–7 shows step 2 of the Connector Management Wizard:

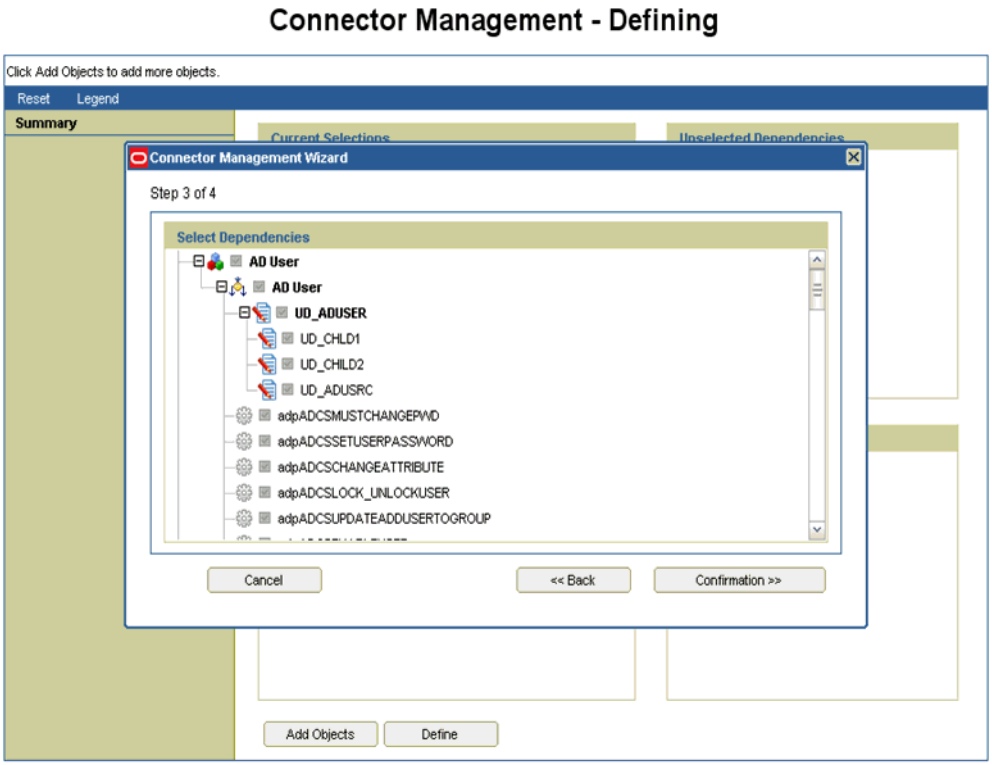
Figure 11–7 Step 2 of the Connector Management Wizard



8. After you review the list of objects that you have selected, click **Confirmation**.

   Figure 11–8 shows step 3 of the Connector Management Wizard with the list of selected connector objects:

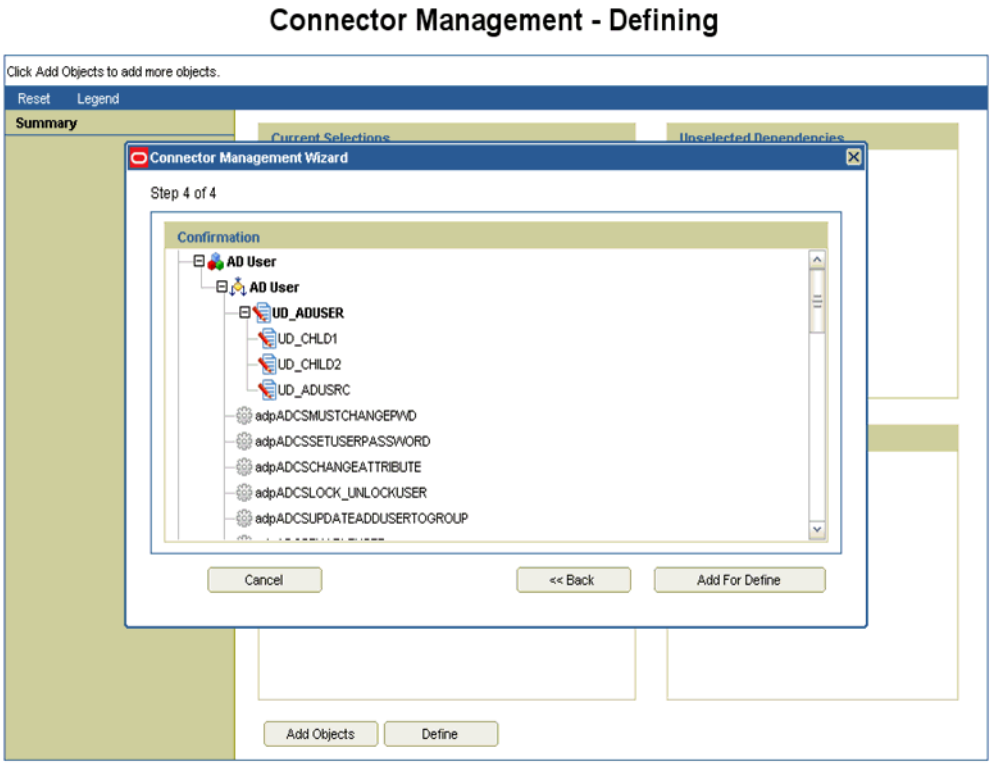
Figure 11–8 Step 3 of the Connector Management Wizard



9. Click **Add For Define**.

   Figure 11–9 shows step 4 of the Connector Management Wizard:

Figure 11–9 Step 4 of the Connector Management Wizard

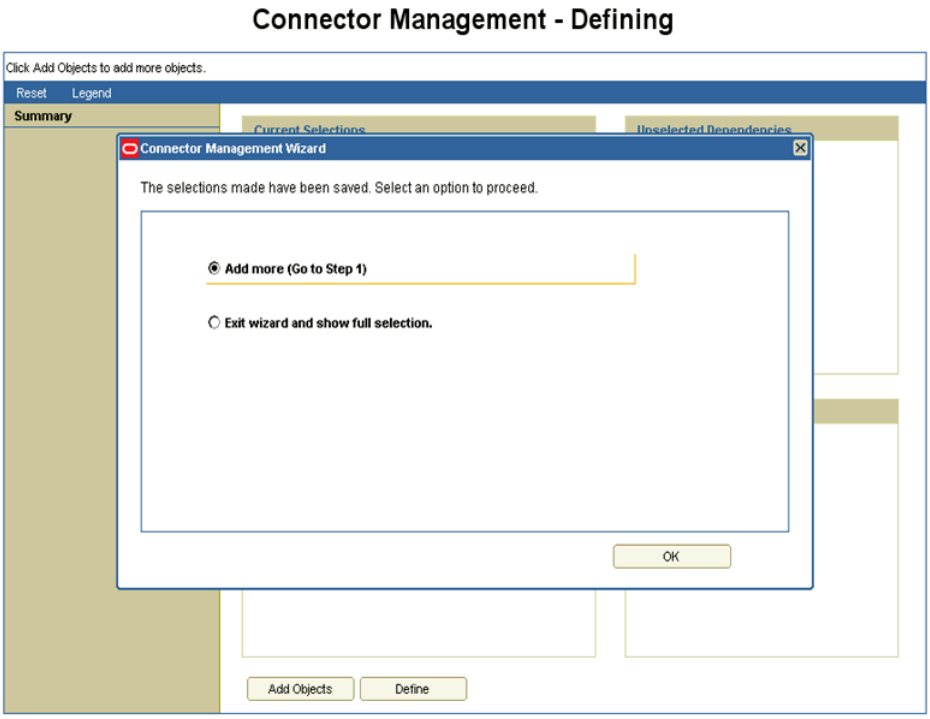


10. To proceed, select any one of the following options, and click OK:

    - **Add more (Go to Step 1):** Select this option if you want to go to step 1 of the Connector Management Wizard and select more connector objects.
    - **Exit wizard and show full selection:** Select this option if you want to exit the Connector Management Wizard and display the complete list of selected connector objects.

    Figure 11–10 shows the page with the options to add more connector objects or to exit the wizard:

Figure 11–10 Options to Select More Objects or Exit



11. On the page that is displayed, only objects shown in the Current Selections list are included in the connector definition. You can drag objects across lists. For example, you can drag an adapter from the Current Selections list to the Unselected Children list. After you make the required changes, click **Define**.

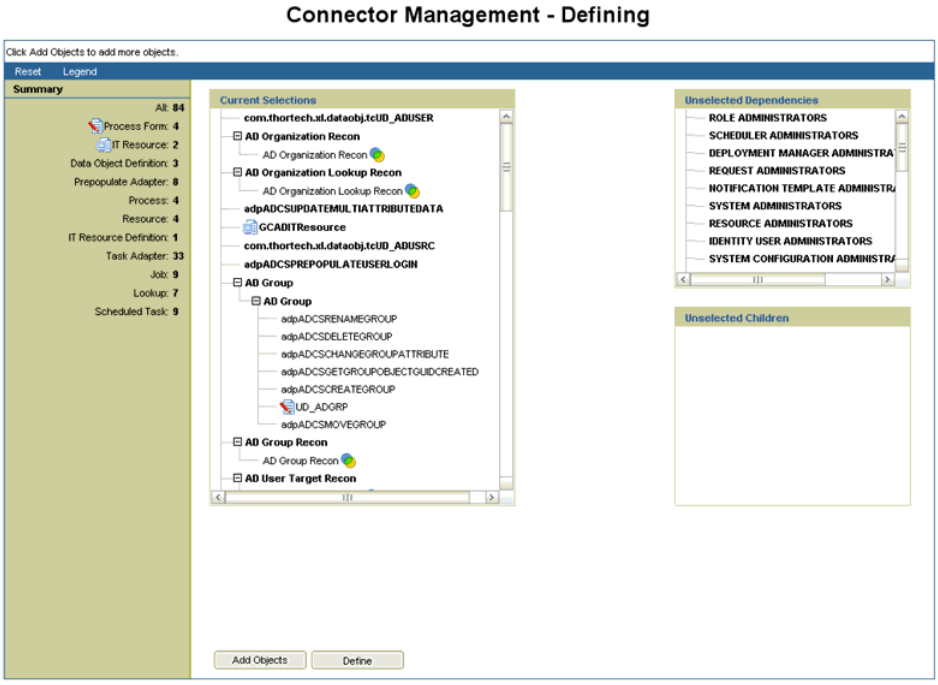
Note: Make sure that you have added all the Oracle Identity Manager connector objects specific to defining connector. If you do not have a specific connector object while defining the connector, then upgrade, clone, or uninstall may not handle the undefined object.

The following are Oracle Identity Manager artifacts that are generally associated with almost all the connectors:

- Resource objects
- Event handlers
- Process forms
- IT resources
- Data object definitions
- Prepopulate adapters
- Processes
- IT resource type definitions
- Task adapters
- Lookups
- Scheduled tasks

Figure 11–11 shows the page with the complete list of selected connector objects that are to be included in the connector definition and the unselected connector dependencies:

Figure 11–11 Selected Connector Objects

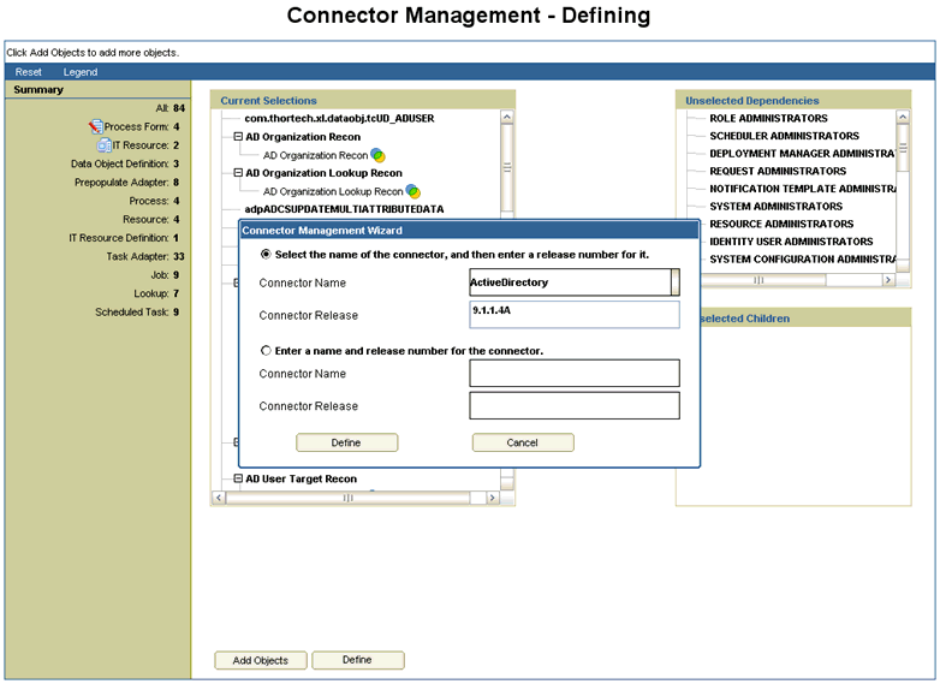


12. In the dialog box that is displayed, select one of the following options:

    - **Select the name of the connector, and then enter a release number for it:** Select this option if an earlier release of this connector already exists on this Oracle Identity Manager installation. In addition, select a connector name and enter a release number.
    - **Enter a name and release number of the connector:** Select this option if an earlier release of this connector does not exist on this Oracle Identity Manager installation. In addition, enter a connector name and release number.

    Figure 11–12 shows the dialog box to specify the connector name and release number:

Figure 11–12 Connector Name and Release Number



13. Click **Define**.
14. At the end of the process, a message stating that the operation was successful is displayed. Click **Close**.

11.6 Cloning Connectors

> **Note:** In this guide, the term **Clone Connectors feature** refers to the set of Oracle Identity Self Service pages that you can use to clone connectors.

This section describes the procedure to create a copy of a connector by setting new names for some of the objects that comprise the connector. The outcome of the process is a new connector XML file. Most of the connector objects, such as Resource Object, Process Definition, Process Form, IT Resource Type Definition, IT Resource Instances, Lookup Definitions, Adapters, Reconciliation Rules and so on in the new connector XML file have new names.

Note: Oracle Identity Manager offers a different feature for using a single connector to integrate:

- Multiple installations of a particular target system with Oracle Identity Manager
- A target system that stores data about multiple user types (for example, employee and contractor) and requires Oracle Identity Manager to provide a different resource object for each user type

See the connector guide for information about how to use access policies to create resource objects for different user types on a particular target system.

This section contains the following topics:

- Guidelines for Cloning a Connector
- Cloning a Connector
- Postcloning Steps

11.6.1 Guidelines for Cloning a Connector

Apply the following guidelines while using the Clone Connectors feature:

- A connector must be compatible with the Clone Connectors feature before you can use the utility to create a clone of the connector. For an Oracle-released connector, see the connector guide for information about whether or not the connector is supported by the Clone Connectors feature.
- Validation performed on the names of connector objects does not cover the names of objects that belong to other connectors. However, when you import the connector XML file that is created by the Clone Connectors feature, the Deployment Manager throws an error when it encounters duplicate object names. This is illustrated by the following example:

  AD USER is the name of a resource object belonging to the Microsoft Active Directory connector. Suppose My_RO is the name of an existing resource object defined in the Oracle Identity Manager database. If the new name that you specify for the AD_USER resource object is My_RO, then the Clone Connectors feature does not display an error message stating that a resource object with the specified name already exists.

11.6.2 Cloning a Connector

Cloning a connector involves performing a two-step procedure:

- Step 1: Create the connector XML file for the cloned connector
- Step 2: Install the clone connector

Step 1: Create the connector XML file for the cloned connector

To create the connector XML file for the cloned connector:

1. Log in to Oracle Identity System Administration.
2. In the left pane, under Provisioning Configuration, click **Manage Connector**.
3. The next step depends on the source XML that you want to use to create the clone:

- If you want to use a connector XML file as the source, then:
  a. Click **Clone** in the upper-right corner.
  b. On the Step 1: XML Selection from File System page, use the Browse option to navigate to and select the connector XML file.
  c. Click **Continue**.
- If you want to use the connector XML that was stored in the database when the connector was defined, then:
  a. Use the Search feature to search for the connector.
  b. In the search results that are displayed, click the Clone icon in the row for the connector that you want to clone.

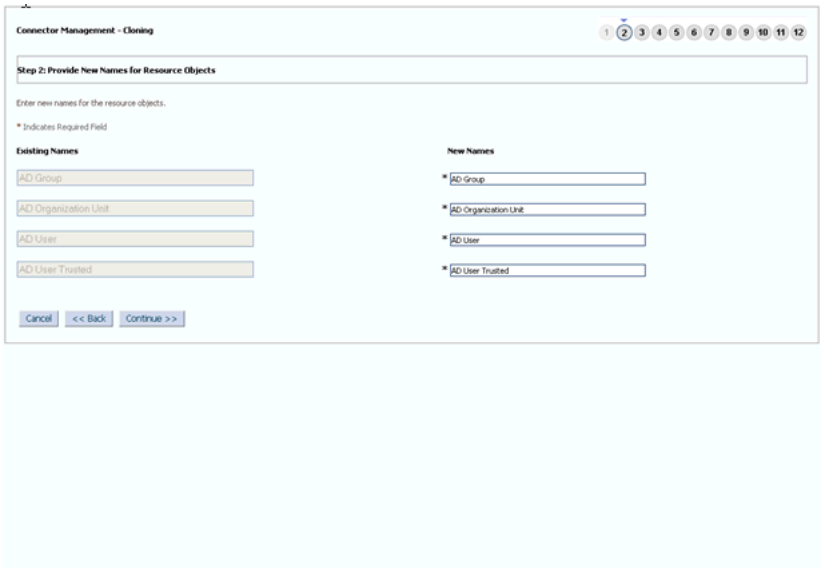
4. On the Step 2: Provide New Names for ROs page, enter new names for the resource objects of the clone.

   If the connector has multiple resource objects, then the new name that you specify for each resource object must be different from the names of all the existing resource objects of that connector.

   Click **Continue** after you specify new names for all the resource objects.

   Figure 11–13 shows the Provide New Names for Resource Objects page of the Connector Management - Cloning wizard:

Figure 11–13 The Provide New Names for Resource Objects Page



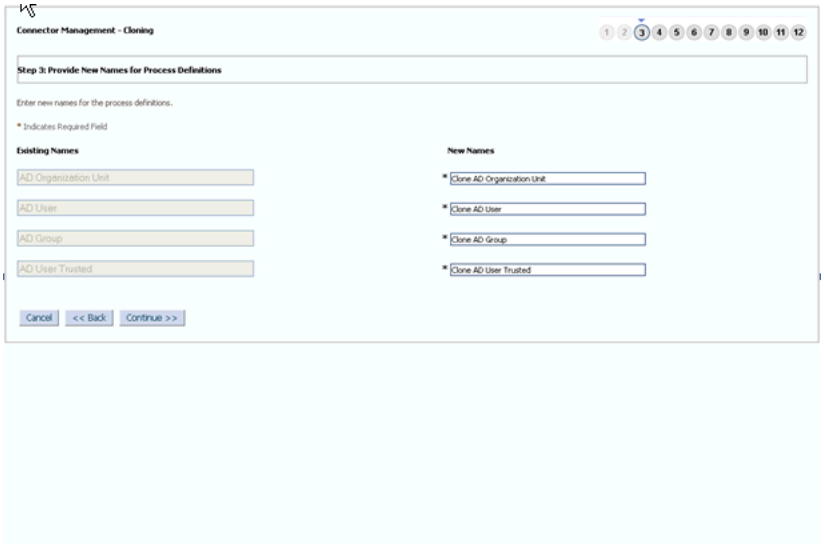
5. On the Step 3: Provide New Names for Process Definitions page, enter new names for the process definitions of the clone.

   If the connector has multiple process definitions, then the new name that you specify for each process definition must be different from the names of all the existing process definitions of that connector.

   Click **Continue** after you specify new names for all the process definitions.

   Figure 11–14 shows the Provide New Names for Process Definitions page of the Connector Management - Cloning wizard:

Figure 11–14 The Provide New Names for Process Definitions Page



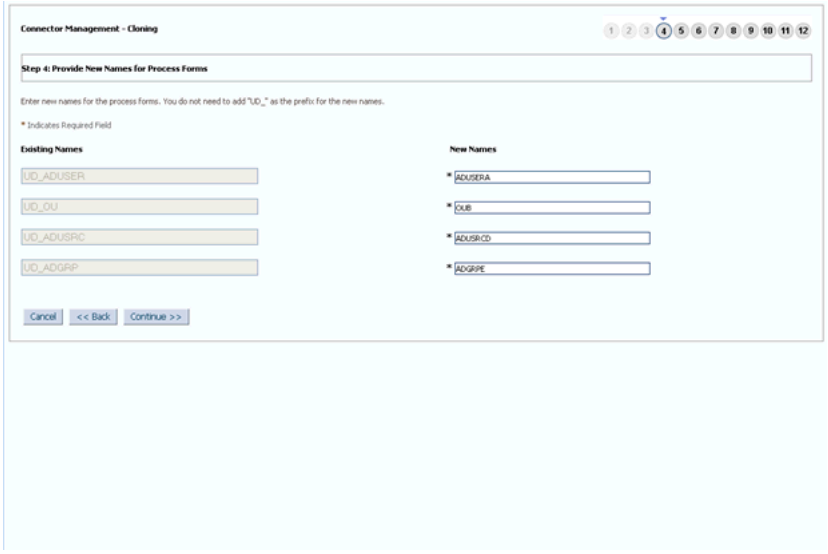
6. On the Step 4: Provide New Names for Process Forms page, enter new names for the process forms of the clone.

   If the connector has multiple process forms, then the new name that you specify for each process form must be different from the names of all the existing process forms of that connector.

   Click **Continue** after you specify new names for all the process forms.

   Figure 11–15 shows the Provide New Names for Process Forms page of the Connector Management - Cloning wizard:

Figure 11–15 The Provide New Names for Process Forms Page



7. On the Step 5: Provide New Names for IT Resource Type Definitions page, enter new names for the IT resource type definitions of the clone.

If the connector has multiple IT resource type definitions, then the new name that you specify for each IT resource type definition must be different from the names of all the existing IT resource type definitions of that connector.

Click **Continue** after you specify new names for all the IT resource type definitions.

Figure 11–16 shows the Provide New Names for IT Resource Type Definitions page of the Connector Management - Cloning wizard:

Figure 11–16 The Provide New Names for IT Resource Type Definitions Page

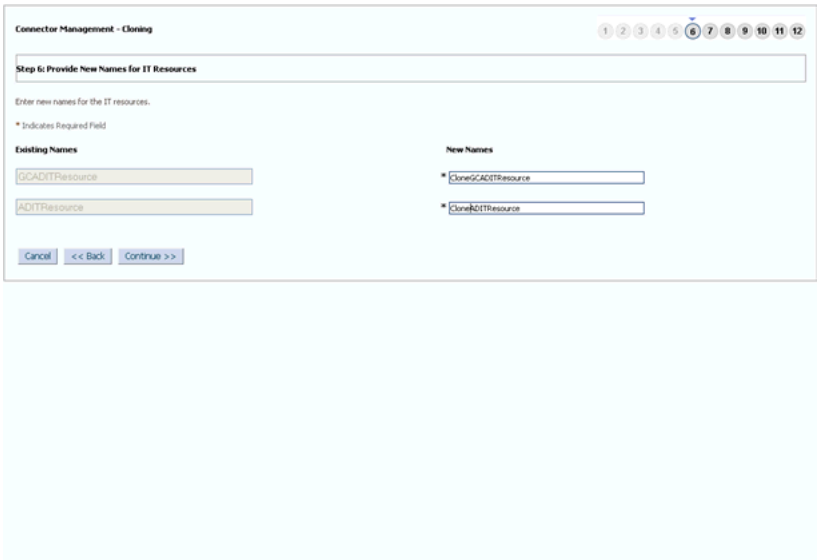
8. On the Step 6: Provide New Names for IT Resources page, enter new names for the IT resources of the clone.

   If the connector has multiple IT resources, then the new name that you specify for each IT resource must be different from the names of all the existing IT resources of that connector.

   Click **Continue** after you specify new names for all the IT resources.

   Figure 11–17 shows the Provide New Names for IT Resource Type Definitions page of the Connector Management - Cloning wizard:

Figure 11–17 The Provide New Names for IT Resources Page



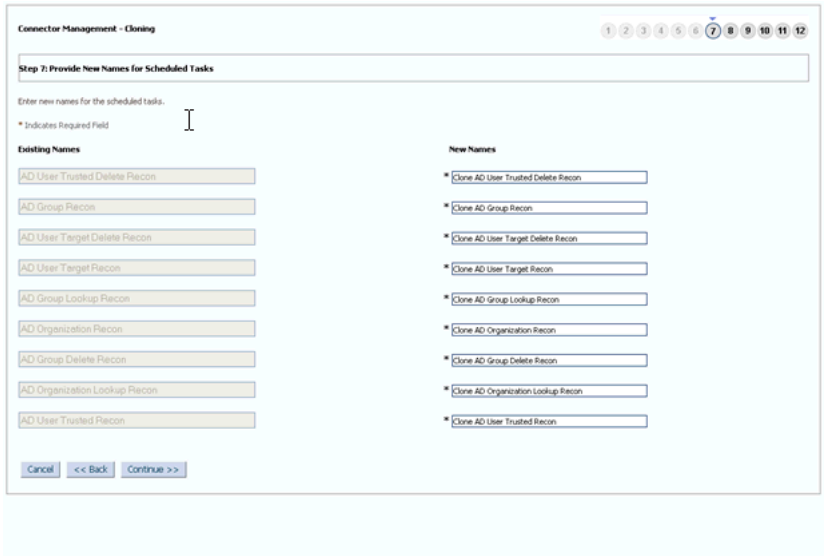
9. On the Step 7: Provide New Names for Scheduled Tasks page, enter new names for the scheduled tasks of the clone.

   Enter new names for the scheduled tasks. However, you cannot use the same set of scheduled tasks for the clone and the original connector.

   Click **Continue**.

   Figure 11–18 shows the Provide New Names for Scheduled Tasks page of the Connector Management - Cloning wizard:

Figure 11–18 The Provide New Names for Scheduled Tasks Page



10. On the Step 8: Provide New Names for Scheduled Jobs page, enter new names for the scheduled jobs of the clone.

    Click **Continue**.

11. On the Step 9: Provide New Names for Lookup Type Definitions page, enter new names for the lookup definitions of the clone.

    Click **Continue**.

    Figure 11–19 shows the Provide New Names for Lookup Type Definitions page of the Connector Management - Cloning wizard:

Figure 11–19 The Provide New Names for Lookup Type Definitions Page

12. On the Step 10: Provide a Prefix for Adapters page, enter the string that is set as the prefix for the copies of the adapters. Then, click **Continue**.

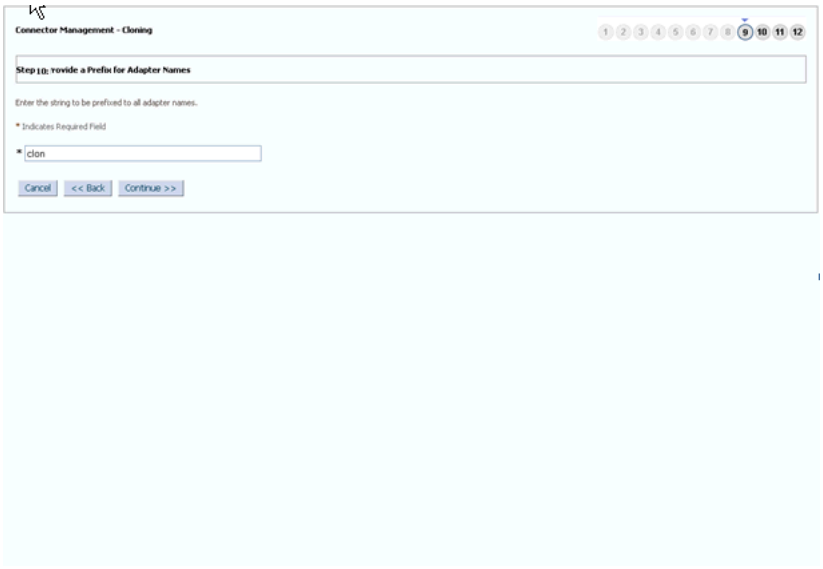
    You must ensure that the prefix that you specify does not cause the full name of any adapter to exceed 80 characters. The Clone Connectors feature cannot check if this limit is exceeded. However, when you import the connector XML file created

for the clone, the Deployment Manager throws an error. Remember that the Deployment Manager is called even when you build a deployment package for the clone and use the Install Connectors feature to install the clone.

You can use the Design Console to determine the character length of the longest adapter name.

Figure 11–20 shows the Provide a Prefix for Adapters page of the Connector Management - Cloning wizard:

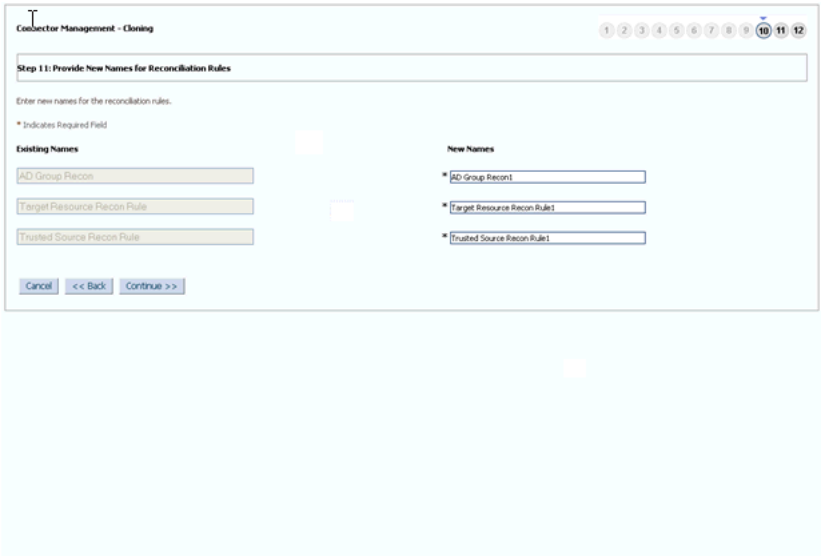
Figure 11–20 The Provide a Prefix for Adapters Page



13. On the Step 11: Provide New Names for Reconciliation Rules page, enter new names for the reconciliation rules of the clone.

    Figure 11–21 shows the Provide New Names for Reconciliation Rules page of the Connector Management - Cloning wizard:

Figure 11–21 The Provide New Names for Reconciliation Rules Page



14. On the Step 12: Object Names Summary page, review the names that you have set for the connector objects of the clone and then click **Confirm**.

    Figure 11–22 shows the Object Names Summary page of the Connector Management - Cloning wizard:

Figure 11–22 The Object Names Summary Page

Connector Management - Cloning

1 2 3 4 5 6 7 8 9 10 **11** **12**

Step 12: Object Names Summary

Review the new object names, and then click Confirm to proceed with the cloning operation.

Resource Objects mapping summary.

Existing Object Names	New Object Names
AD Group	Clone AD Group
AD Organization Unit	Clone AD Organization Unit
AD User	Clone AD User
AD User Trusted	Clone AD User Trusted

Process Definition mapping summary.

Existing Object Names	New Object Names
AD Organization Unit	Clone AD Organization Unit
AD User	Clone AD User
AD Group	Clone AD Group
AD User Trusted	Clone AD User Trusted

Process Form mapping summary.

Existing Object Names	New Object Names
UD_ADUSER	UD_ADUSERA
UD_OU	UD_OUB
UD_ADUSRC	UD_ADUSRCD
UD_ADGRP	UD_ADGRPE

IT Resource type definition mapping summary.

Existing Object Names	New Object Names
AD Server	Clone AD Server

IT Resource mapping summary.

Existing Object Names	New Object Names
GCADITResource	CloneGCADITResource
ADITResource	CloneADITResource

Scheduled tasks mapping summary.

Existing Object Names	New Object Names
AD User Trusted Delete Recon	Clone AD User Trusted Delete Recon
AD Group Recon	Clone AD Group Recon
AD User Target Delete Recon	Clone AD User Target Delete Recon
AD User Target Recon	Clone AD User Target Recon
AD Group Lookup Recon	Clone AD Group Lookup Recon
AD Organization Recon	Clone AD Organization Recon
AD Group Delete Recon	Clone AD Group Delete Recon
AD Organization Lookup Recon	Clone AD Organization Lookup Recon
AD User Trusted Recon	Clone AD User Trusted Recon

Lookup definitions mapping summary.

Existing Object Names	New Object Names
Lookup.AD Group Type	Lookup.CloneAD Group Type
Lookup.AD.FieldsForValidation	Lookup.CloneAD.FieldsForValidation
AtMap.ADAMGroup	AtMap.CloneADAMGroup
Lookup.ADReconciliation.TransformationMap	Lookup.CloneADReconciliation.TransformationMap
AtMap.AD.RemoteScriptlookUp	AtMap.CloneAD.RemoteScriptlookUp
Lookup.AD.BLOBAttribute.Values	Lookup.CloneAD.BLOBAttribute.Values
Lookup.ADAMReconciliation.FieldMap	LookupClone.ADAMReconciliation.FieldMap
AtMap.ADAM	AtMap.CloneADAM
Atmap.RM	Atmap.CloneRM
Lookup.AD.Constants	Lookup.CloneAD.Constants
Lookup.ADReconciliation.Organization	Lookup.CloneADReconciliation.Organization
Lookup.AD.Country	Lookup.CloneAD.Country
AtMap.ADGroup	AtMap.CloneADGroup
Lookup.ADReconciliation.FieldMap	Lookup.CloneADReconciliation.FieldMap
lookup.AD.GroupChildData	lookup.CloneAD.GroupChildData
Lookup.ADReconciliation.GroupLookup	Lookup.CloneADReconciliation.GroupLookup
AtMap.AD	AtMap.CloneAD
Lookup.AD.Domains	Lookup.CloneAD.Domains
Lookup.ADGroupReconciliation.FieldMap	Lookup.CloneADGroupReconciliation.FieldMap
Lookup.ADAMGroupReconciliation.FieldMap	Lookup.CloneADAMGroupReconciliation.FieldMap
Lookup.AD.Configuration	Lookup.CloneAD.Configuration

Adapter names mapping summary.

Existing Object Names	New Object Names
ADCS Update Multi Attribute Data	ClonADCS Update Multi Attribute Data
ADCS Prepopulate User Last Name	ClonADCS Prepopulate User Last Name
ADCS Execute Remote Script	ClonADCS Execute Remote Script
ADCS Change Org Name	ClonADCS Change Org Name
ADCS Prepopulate User Password	ClonADCS Prepopulate User Password
ADCS Prepopulate User Middle Name	ClonADCS Prepopulate User Middle Name
ADCS Rename Group	ClonADCS Rename Group
ADCS Must Change PWD	ClonADCS Must Change PWD
ADCS Lock_Unlock User	ClonADCS Lock_Unlock User
ADCS Remove User From Group	ClonADCS Remove User From Group
ADCS Add User To Group	ClonADCS Add User To Group
ADCS Prepopulate UserPrincipalName	ClonADCS Prepopulate UserPrincipalName
ADCS Rename User Account	ClonADCS Rename User Account
ADCS Get Group ObjectGUID Created	ClonADCS Get Group ObjectGUID Created
ADCS Prepopulate AD Group Name	ClonADCS Prepopulate AD Group Name
ADCS Remove Multi Attribute Data	ClonADCS Remove Multi Attribute Data
ADCS Get USNChanged	ClonADCS Get USNChanged
ADCS Pwd Never Expires	ClonADCS Pwd Never Expires
ADCS Set User Password	ClonADCS Set User Password
ADCS Get USNCreated	ClonADCS Get USNCreated
ADCS Update Redirect Mail ID	ClonADCS Update Redirect Mail ID
ADCS Check Process Parent Org	ClonADCS Check Process Parent Org
ADCS Add Multi Attribute Data	ClonADCS Add Multi Attribute Data
ADCS Create Group	ClonADCS Create Group
ADCS Set Account Exp Date	ClonADCS Set Account Exp Date
ADCS Disable User	ClonADCS Disable User
ADCS Move User	ClonADCS Move User
ADCS Create OU	ClonADCS Create OU
ADCS Change Attribute	ClonADCS Change Attribute
ADCS Delete OU	ClonADCS Delete OU
ADCS Create User	ClonADCS Create User
ADCS Prepopulate User Full Name	ClonADCS Prepopulate User Full Name
ADCS Prepopulate User First Name	ClonADCS Prepopulate User First Name
ADCS Delete User	ClonADCS Delete User
ADCS Update Add User to Group	ClonADCS Update Add User to Group
ADCS Move Group	ClonADCS Move Group
ADCS Move OU	ClonADCS Move OU
ADCS Prepopulate User Login	ClonADCS Prepopulate User Login
ADCS Delete Group	ClonADCS Delete Group
ADCS Change Group Attribute	ClonADCS Change Group Attribute
ADCS Enable User	ClonADCS Enable User

Reconciliation rules mapping summary.

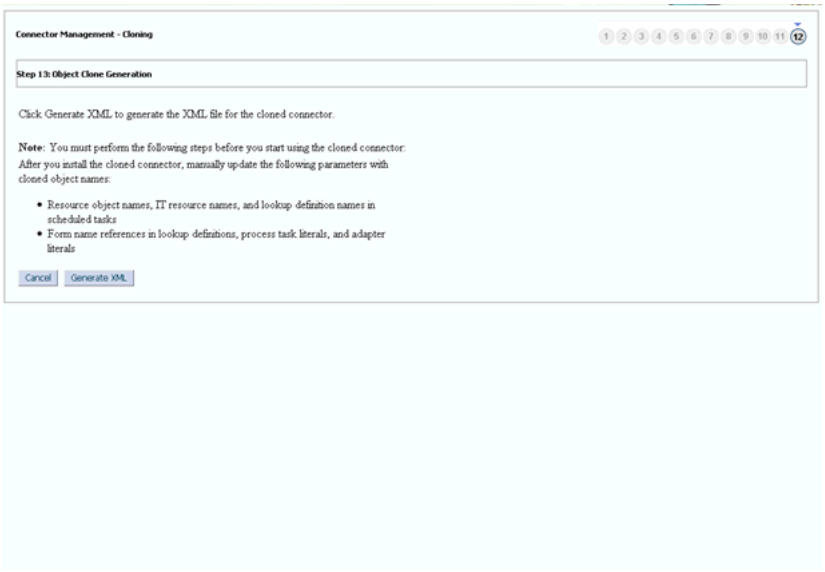
Existing Object Names	New Object Names
AD Group Recon	AD Group Recon1
Target Resource Recon Rule	Target Resource Recon Rule1
Trusted Source Recon Rule	Trusted Source Recon Rule1

Cancel << Back Confirm

15. On the Step 13: Object Clone Generation page, click **Generate XML**.

    Figure 11–23 shows the Object Clone Generation page of the Connector Management - Cloning wizard:

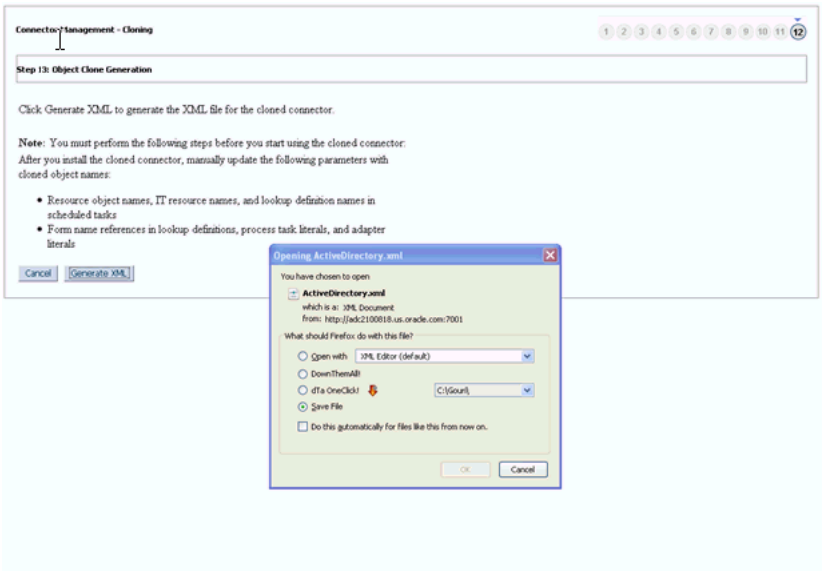
Figure 11–23 The Object Clone Generation Page



16. In the File Download dialog box, use the Save option to save the connector XML file of the clone to a location of your choice.

    Figure 11–24 shows the File Download dialog box:

Figure 11–24 The File Download Dialog Box



Step 2: Install the clone connector

You can install the clone connector by using one of the following approaches:

Note: You can install the clone connector on either the same or a different Oracle Identity Manager installation.

- Use the Deployment Manager to import the connector XML file. If you use Deployment Manager import to install the connector, then you need to define the cloned connector. This will enlist the cloned connector in the list of connectors in Connector Management Search. If the connector is imported in different Oracle Identity Manager environment where the original connector does not exist, then you need to upload the related Jar files of the connector using JarUpload utility and adapters need to be compiled after all connector jars have been uploaded.
- Create a deployment package for the cloned connector, and then install it using the Install Connectors feature. For a sample, see the contents of the deployment package for any Oracle-released connector.

11.6.3 Postcloning Steps

After a copy of the connector is created by setting new names for connector objects, some objects might contain the details of the old connector objects. Therefore, you must modify the following Oracle Identity Manager objects to replace the base connector artifacts or attribute references with the corresponding cloned artifacts or attributes:

- **Lookup Definition:** If the lookup definition contains the old lookup definition details, then it must be modified to provide the new cloned lookup definition names. If the encode and decode values are referring the base connector attribute references, then these must be replaced with new cloned attributes.
- **Scheduled Task:** The base connector resource object name in the scheduled task must be replaced with the cloned resource object name. If the scheduled task parameter has any data referring to the base connector artifacts or attributes, then these must be replaced with the new cloned connector artifacts or attributes.

11.7 Exporting Connector Object Definitions in Connector XML Format

As mentioned earlier, the Oracle Identity Manager database stores the definitions of all connector objects. You can export these definitions to create a connector XML file for a particular connector. By using the Deployment Manager, you can import the connector XML file to create the connector object definitions in another Oracle Identity Manager installation.

Alternatively, you can use the connector XML file as one of the components of a deployment package that you create for the connector. This deployment package can then be installed using the Install Connectors feature. For a sample, see the contents of the deployment package for any Oracle-released connector. Another important component of a deployment package is the configuration XML file, which is used by the Install Connectors feature. You must manually create the configuration XML file.

See Also : Connector guide for information about the contents of the configuration XML file

To export connector object definitions in connector XML format:

1. Log in to Oracle Identity System Administration.
2. In the left pane, under Provisioning Configuration, click **Manage Connector**.

3. You can use one of the following options to export the connector XML file:
   - If you want the XML file to include definitions of only specific connector objects, then use the Export button to open the Deployment Manager. See the "Using the Deployment Manager" chapter in the connector guide for detailed information about using this feature to select connector objects whose definitions you want to include in the connector XML file.
   - If you want to create the connector XML file out of the connector XML stored in the database when the connector was defined, then:
     a. In the Connector Management page, use the Search feature to display the connector for which you want to create the connector XML file.
     b. Use the Export icon displayed in the connector row to export the connector XML file from the entry created in the database when defining the connector.

11.8 Upgrading Connectors

Connector Upgrade utility is responsible for upgrading the Oracle Identity Manager artifacts from source version to the target version, by retaining the customer customization done on the source connector. Connector upgrade does not handle connector library upgrade/update. User need to manually upgrade the libraries involved in connector.

The following are sample scenarios that describe a need for upgrading a connector:

- Reconfiguring or customizing an existing connector

  After you install a connector, you might customize or reconfigure it according to your requirements. For example, you might add new attributes for reconciliation and provisioning and modify the scheduled tasks for reconciliation or lookup field synchronization. Ideally, you would make these changes to the connector on a staging server. You would then want to upgrade the connector deployed on your production server to the version that you create by making changes on the staging server.

- Upgrading a customer-developed connector

  You might have developed your own connector. When an Oracle-released upgrade is available for your connector, you might want to upgrade from your connector to the Oracle-released connector. For example, suppose you have developed and are using a connector for IBM Lotus Notes and Domino. When Oracle ships a new release of Oracle Identity Manager Connector for IBM Lotus Notes and Domino, you might want to use some of the features included in the new release. You can use the Upgrade Connectors feature to upgrade from your connector to the Oracle-released connector.

- Upgrading an Oracle-released connector

  Oracle ships connector upgrades. An upgrade includes enhancements and fixes that you might need. For example, if you are currently using SAP User Management release 9.1.2, then you might want to upgrade to release 9.1.2.3 of the same connector when that release is available.

In scenarios such as these, you can use the Upgrade Connectors feature to upgrade the connector.

Upgrading connectors can be done by two ways:

- Silent mode upgrade: Used in staging and production environments

- Wizard mode upgrade: Used in development environment

In this guide, Wizard upgrade, which is performed using Oracle Identity System Administration pages is described.

This section is divided into the following topics:

- Upgrade Use Cases Supported by the Connector Upgrade Feature
- Summary of the Upgrade Procedure
- Procedure to Upgrade a Connector

11.8.1 Upgrade Use Cases Supported by the Connector Upgrade Feature

The following types of source connectors are supported by the Upgrade Connectors feature:

- Customer-developed connectors
- Oracle-released connectors that are not supported by the Install Connectors feature
- Oracle-released connectors that are supported by the Install Connectors feature
- Oracle-released connectors that are supported by the Install Connectors feature and have been customized
- Cloned connectors

The upgrade process does not cover the following objects:

- E-mail definitions
- Password policies
- Error message definitions
- Business rule definitions
- Object forms
- Access policies

Note:

- Connector lifecycle management does not support the upgrade of a trusted connector if the source connector uses the Xellerate User resource object for trusted source configuration. Therefore, you must manually upgrade the connector. Contact Oracle Support for more information.
- Connector lifecycle management does not support the upgrade of a connector from the target mode (source version) to the trusted mode (target version). Similarly, upgrading from trusted mode to the target mode is also not supported.

Use Case 1: Custom-Developed Source Connector

A custom-developed source connector must meet the following requirements so that it is compatible with the Upgrade Connectors feature:

- The connector must be defined in Oracle Identity Manager. See "Defining Connectors" on page 11-12 if you want to manually define the connector.

- The connector must have a configuration XML file. See the connector guide for information about configuration XML files.

The following are sample events that can take place before you upgrade a custom-developed source connector:

- You develop the connector and its configuration XML file.
- Create a deployment package that is compatible with the Connector Installation feature. When you use this feature to deploy the connector on the production server, the connector is automatically defined at the end of the installation process.
- You use the connector for reconciliation and provisioning. Target system resources are allocated (through reconciliation and provisioning) for Oracle Identity Manager Users.
- You modify the connector on the staging server, redefine it, and then regenerate the connector XML file.

Use Case 2: Oracle-released connector that is not supported by the Install Connectors feature

A connector that is not supported by the Install Connectors feature connector must meet the following requirements so that it is compatible with the Upgrade Connectors feature:

- The connector must be defined in Oracle Identity Manager. See "Defining Connectors" on page 11-12 if you want to manually define the connector.
- The connector must have a configuration XML file. See the connector guide for information about configuration XML files.

Sample events and the upgrade procedure for this use case are the same as those for Use Case 1.

Use Case 3: Oracle-released connector that is installed using the Install Connectors feature

A connector that is installed using the Install Connectors feature meets the requirements specified for Use Cases 1 and 2.

Use Case 4: Oracle-released connector that has been installed and then customized

A connector that is supported by the Install Connectors feature meets the requirements specified for Use Cases 1 and 2. However, customizations are overwritten during the upgrade process. For example, if you have added an attribute in a scheduled task and also modified the JAR file for reconciliation, then this customization would be lost after the upgrade. To work around this issue:

1. Keep a record of customizations that you implement on a connector.
2. After you upgrade the connector, reapply the customizations.

Use Case 5: Cloned connector

A connector that is installed using the Clone Connectors feature meets the requirements specified for Use Cases 1 and 2.

After the upgrade operation, you can use each clone to manage resource data that was collected through the clone before the upgrade.

11.8.2 Connector Object Changes Supported by the Upgrade Connectors Feature

Before you upgrade a connector, you might have reconfigured or customized the connector by making changes in individual connector objects. The upgrade process itself changes individual connector objects. The following sections list connector object changes supported by the Upgrade Connectors feature. These changes may have been performed manually (that is, at any time before the Upgrade Connectors feature is used) or may be performed by the Upgrade Connectors feature itself.

- Resource Object Changes
- Process Definition Changes
- Process Form Changes
- Lookup Definition Changes
- Adapter Changes
- Rule Changes
- IT Resource Type Changes
- IT Resource Changes
- Scheduled Task Changes

11.8.2.1 Resource Object Changes

The Upgrade Connectors feature can run on a resource object on which any combination of the following changes have been made. In addition, an upgrade operation might involve any combination of the following changes to a resource object.

- Status definitions can be added or deleted.
- Administrators can be assigned or deleted.
- Password policies can be added or deleted.
- User-defined fields (UDFs) can be added or deleted.
- Dependencies with other resource objects can be assigned or deleted.
- Object authorizers can be assigned or deleted. In addition, the priority number assigned to the authorizers can be modified.
- Process determination rules can be assigned or deleted.
- Event-handler adapters can be assigned or deleted.
- Resource object fields that are not present in the connector XML of the target connector are marked as obsolete.
- Customizations performed on the resource object are not retained.

After the upgrade, the new name of the resource object is the one specified in the connector XML of the target connector.

11.8.2.2 Process Definition Changes

The Upgrade Connectors feature can run on a process definition on which any combination of the following changes have been made. In addition, an upgrade operation might involve any combination of the following changes to a process definition.

- The existing process definition can be replaced by a new process definition.

- The existing provisioning definition can be renamed.
- Existing reconciliation field mappings can be retained without change or modified.
- New process tasks can be added.
- Custom process tasks can be retained without a change.
- Default process tasks can be retained, but you need to confirm that there are no changes in the default process task in the new version. Refer to the connector guide for more information.
- Any combination of the following changes can be made to an existing process task:
  - The name and properties of the task can be modified.
  - An attached event handler-adapter can be modified.
  - Preceding and dependent tasks can be added, modified, or deleted.
  - New response codes can be added.
  - Existing response codes can be modified or deleted.
  - New tasks can be generated.
  - Undo tasks and recovery tasks can be modified.
  - Task-to-object status mapping can be modified.
  - Assignment rules can be modified.
- Existing process tasks can be deleted.

After the upgrade, the new name of the process definition is the one specified in the connector XML of the target connector.

11.8.2.3 Resource Bundle Changes

To update the resource bundles:

1. If there are any customization on the resource bundles such as adding new entries to the connector resource bundles, the changes need to be applied on the resource bundles present in the "resources" folder of the connector distribution bundle. The existing resource bundles present in Oracle Identity Manager database can be downloaded using the DownloadResourceBundles utility available under *OIM_HOME*/server/bin.
2. Use DownloadResourceBundles utility (available under *OIM_HOME*/server/bin) to delete all the resource bundles specific to the connector from Oracle Identity Manager database.
3. Use UploadResourceBundles utility (available under *OIM_HOME*/server/bin) to upload all the resource bundles specific to the connector to Oracle Identity Manager database.

11.8.2.4 Process Form Changes

The Upgrade Connectors feature can run on a process form on which any combination of the following changes have been performed. In addition, an upgrade operation might involve any combination of the following changes to a process form.

Note:

- An upgrade operation works on only the active version of the process form. No changes are made to earlier versions.
- The existing process form cannot be renamed.

- Columns can be added, modified, or deleted.
- Child forms can be added, modified, or deleted.
- Pre-populate adapters can be added.
- The name, mappings, order, and rule of existing pre-populate adapters can be modified.
- The user can manually add the customizations to the active version if they wish to add certain fields to the new version that were present in the existing form.
- If the form attribute is retained and the corresponding connector objects, for example Lookup Definition and IT Resource Type Definition are removed to which this attribute has references, then you need to modify the form attribute properties by pointing it to the correct connector object.

After the upgrade, the name of the process form is the version number of the upgraded connector.

11.8.2.5 Lookup Definition Changes

The Upgrade Connectors feature can run on a lookup definition on which any combination of the following changes have been made. In addition, an upgrade operation might involve any combination of the following changes to a lookup definition.

- Lookup definitions can be added.

Note: Existing lookup definitions are not deleted during an upgrade operation.

- Existing lookup definitions can be retained or modified. During an upgrade operation, new entries in an existing lookup definition are appended after the existing entries.

11.8.2.6 Adapter Changes

The Upgrade Connectors feature can run on an adapter on which any combination of the following changes have been made. In addition, an upgrade operation might involve any combination of the following changes to an adapter.

Note: Existing adapters are not deleted during an upgrade operation.

- New adapters can be added.
- The custom adapters are retained as part of upgrade. If there are any customization on the default adapters, these changes need to be applied after upgrade as all the default adapters is overwritten.

- After applying the customization on the default adapters (if there are any), the corresponding mapping for these adapters in Process Task, form field, and data object manager need to be verified for mapping.

11.8.2.7 Rule Changes

The Upgrade Connectors feature can run on a rule on which any combination of the following changes have been made. In addition, an upgrade operation might involve any combination of the following changes to a rule.

- New rules can be added.
- If there are any customizations in default Rules, these customizations need to be applied after the upgrade as all default Rules is overwritten.

11.8.2.8 IT Resource Type Changes

The Upgrade Connectors feature can run on an IT resource type on which any combination of the following changes have been made. In addition, an upgrade operation might involve any combination of the following changes to an IT resource type.

- The existing IT resource type can be replaced by a new IT resource type.
- In an existing IT resource type, new parameters can be added and existing parameters can have their default values and types modified or deleted.
- All custom parameters are displayed while mapping IT Resource Type definitions. You can retain the custom parameters.

11.8.2.9 IT Resource Changes

The Upgrade Connectors feature can run on an IT resource on which any combination of the following changes have been made. In addition, an upgrade operation might involve any combination of the following changes to an IT resource.

- The parameter retained for IT Resource Type definition is available for all the IT Resource instances of this type. If an existing parameter in IT Resource Type definition is not retained, then this parameter will not available in all the IT Resource instances of this type.
- In an existing IT resource, new parameters can be added and existing parameters can have their default values and types modified or deleted.

After the upgrade, the new name of the IT Resource Type definition is the one specified in the connector XML of the target connector.

11.8.2.10 Scheduled Task Changes

The Upgrade Connectors feature can run on a scheduled task that has been retained or existing scheduled tasks have been replaced by new scheduled tasks.

11.8.3 What Happens When You Upgrade a Connector

See Upgrade Use Cases Supported by the Connector Upgrade Feature for information about the changes that can be put into effect when you upgrade a connector.

In addition, the following events are part of the outcome of an upgrade operation:

- While performing the upgrade procedure, you are prompted to map new connector objects with existing objects. For example, you are prompted to map each resource object in the target connector with a resource object in the source

connector. If the object names are same in both source and target, then for the new object, the corresponding old object need to be mapped. If there are changes in the object names in source and target, then you need to map the object properly by referring the source and target connector release documents. It is your responsibility map the source and target objects properly. If the objects are not mapped properly, then the source object is corrupted by the upgrade process. Therefore, it is mandatory that you must know about all the source and the target connector objects.

11.8.4 Summary of the Upgrade Procedure

The following is a summary of the procedure to upgrade a connector:

Note: The procedure explained in this chapter is based on the best practice in which you first perform the upgrade in a test development environment. All functional use cases need to be tested before applying the upgrade in production server. Wizard mode upgrade should not be used in production, only silent mode need to be used in production server.

---

1. Read through the upgrade procedure.

   This will let you make an estimate of the time for which the connector and, therefore, the target system might be unavailable to Oracle Identity Manager users. You can also determine if you have the Oracle Identity Manager expertise required to complete all the upgrade and post-upgrade steps.

2. Make a note of associations between objects of the source connector and other Oracle Identity Manager objects. For example, make a note of associations between resource objects and access policies.

3. If required, create the connector XML file for a clone of the source connector.

   If the object names in the target connector are different from object names in the source connector, then it is recommended that you first create the connector XML file for the clone connector. "Step 1: Create the connector XML file for the cloned connector" on page 11-22 describes the procedure. While performing the procedure, specify object names that are the same as object names in the target connector. This will help avoid the need for renaming connector objects after you upgrade the connector.

4. Upgrading the source connector to target connector on staging server.

   The XML file contains details of changes to be made to the connector objects of the source connector so that they are converted into the connector objects of the target connector. These changes are applied automatically during the upgrade process.

   To upgrade the source connector:

   a. Back up the Oracle Identity Manager database on the production server.

   b. Perform the steps described in "Preupgrade Procedure" on page 11-42

   c. Perform the steps described in "Silent Mode Upgrade in Staging and Production Environment" on page 11-54 The resulting transformed XML can be generated and used in production server.

5. Use the silent delta XML for connector upgrade.

   To use the delta XML file:

a. Restore the production database on the staging server.

b. Perform the steps described in "Preupgrade Procedure" on page 11-42

c. Perform the steps described in "Silent Mode Upgrade in Staging and Production Environment" on page 11-54

d. Perform the steps described in "Postupgrade Procedure" on page 11-57

6. Verify that the source connector on the production server is the same as the source connector on the staging server. If there are differences in the source connector on the staging server and the production server, then the delta XML file is not correctly imported on the production server.

7. Import the delta XML file on the production server.

   After you verify that the upgraded target connector is working as expected on the staging server, perform the following steps:

   a. Perform the steps described in "Preupgrade Procedure" on page 11-42

   b. Perform the steps described in "Silent Mode Upgrade in Staging and Production Environment" on page 11-54

   c. Perform the steps described in "Postupgrade Procedure" on page 11-57

11.8.5 Procedure to Upgrade a Connector

The following sections discuss the procedure to upgrade a connector:

- "Preupgrade Procedure" on page 11-42
- "Upgrade Procedure" on page 11-43
- "Postupgrade Procedure" on page 11-57

Note: Keep the SOA server running during the upgrade process.

11.8.5.1 Preupgrade Procedure

Before you begin the upgrade procedure, ensure that the following prerequisites are addressed:

- Read through the upgrade procedure documented in this chapter.
- Note down customizations made in the connector objects on source connector.
- Call a Java API to handle workflows that are in progress. See Step 3 of Section 11.8.5.2, "Upgrade Procedure" for information about pending workflows. You need to make sure that there are no requests in pending state for the resource objects that are part of this connector. You also need to complete all the requests before going for connector upgrade. Requests can be closed if they are in a closable state. All the requests associated with the connector resource objects should in one of the following states before starting the upgrade process.
  - Request Completed
  - Request Closed
  - Request Withdrawn
  - Request Failed
  - Request Approval Rejected

  - Operation Approval Rejected
- If required, create the connector XML file for a clone of the source connector.
- Disable all the scheduled tasks.
- Make sure that the connector is defined if there are any customizations done after installing the connector. See "Defining Connectors" on page 11-12 for information about defining connectors.

11.8.5.2 Upgrade Procedure

Upgrading connectors is a two-stage procedure:

- Wizard Mode Upgrade in Staging Environment
- Silent Mode Upgrade in Staging and Production Environment

Wizard Mode Upgrade in Staging Environment

> **Note:** You need to perform preupgrade and post upgrade steps while performing wizard mode upgrade.

To perform the wizard mode upgrade on the staging server:

1. Create a backup of the Oracle Identity Manager database.
2. Create Oracle Identity Manager metadata (MDS) backup. See "Migrating User Modifiable Metadata Files" in the *Developing and Customizing Applications for Oracle Identity Manager* for information about exporting and importing Oracle Identity Manager metadata to and from MDS.
3. Run the connector preupgrade utility.

   A validation script is provided with Oracle Identity Manager. This script performs the following functions:

   - Determines whether the connector that you want to upgrade has been defined in Oracle Identity Manager

     In other words, the script checks whether the connector XML stored in the database when the connector was installed/defined is consistent with the connector object definitions in the database. Apart from checking the consistency of the connector XML, it also checks whether the Connector XML is present in Oracle Identity Manager Database or not. If it is not present, then it displays the corresponding message to define the connector before proceeding with upgrade. Refer the "Defining Connectors" on page 11-12 to perform the procedure to define a connector.
   - Identifies the Oracle Identity Manager scheduled tasks that are currently running.

     You must disable all scheduled tasks that belong to the source connector before you proceed with the upgrade procedure. In addition, it is recommended to disable all other scheduled tasks before proceeding with the upgrade procedure.
   - Identifies the Attestation tasks associated with the resource object of the connector.

     You must complete all the attestation tasks that belong to the source connector before you proceed with the upgrade procedure.

- Identifies all the pending requests associated with the resource objects of the connectors.

  You must either close or complete all the pending requests that belong to the source connector before you proceed with the upgrade procedure.

To run the validation script:

a. Ensure that Oracle Identity Manager is running.

b. In a command window, change to the *OIM_HOME*/server/bin directory.

c. Run the script as follows:

Note: Set *APP_SERVER*, *OIM_ORACLE_HOME*, *JAVA_HOME*, *MW_HOME*, *WL_HOME*, and *DOMAIN_HOME* before running the scripts.

For Unix:

```
sh ConnectorPreUpgradeUtil.sh
```

For Windows:

```
ConnectorPreUpgradeUtil.bat
```

You are prompted to provide the following details:

- Enter Oracle Identity Manager administrator's username: Enter the Oracle Identity Manger administrator's username.
- Enter Oracle Identity Manager administrator's password: Enter the Oracle Identity Manger administrator's password.
- Enter t3 Oracle Identity Manager Server URL: Enter the Oracle Identity Manger server URL. For example, t3://*HOST_NAME*:*HOST_PORT*.
- Enter context factory: Enter the name of the context factory.
- Enter the connector name: Enter the connector name to be validated before upgrade.
- Enter the connector version: Enter the connector version to be validated before upgrade.

On successfully connecting to the Oracle Identity Manager server, a message is displayed.

The output generated by the script is displayed in the command window and is also recorded in the *OIM_HOME*/server/bin/validateUtil.log file.

The action that you must take depends on the message generated by the script:

- If the message states that the connector XML in the database is not consistent with the connector objects defined in the database, then perform the procedure described in the "Defining Connectors" on page 11-12 of the connector guide.
- If the message states that the "connector XML does not exists in Oracle Identity Manager database. Define a connector before upgrade.", then perform the procedure described in the "Defining Connectors" on page 11-12 section of the connector guide before proceeding with upgrade

- If the message contains the names of the scheduled tasks that are currently running, then you must disable all scheduled tasks. To disable a scheduled task, in the Advanced Administration, click **System Management**, search for scheduled jobs, and click the specific scheduled job, and then click **Stop**.
- If the message contains the names of the Attestation Processes of which some attestation tasks associated with the resource object of the connector is pending, then you must complete all the attestation tasks belonging to the connector that you are upgrading before proceeding with the upgrade process.
- If the message contains the names of the pending requests associated with the resource object of the connector, then you must either close or complete all the pending requests belonging to the connector that you are upgrading before proceeding with the upgrade process.

4. Copy the JARs and the resource bundles to the specified directories.

   If the target release also contains new or updated JARs and resource bundles, then download the version of the jar to Oracle Identity Manager, check the version of the jar which is shipped with Oracle Identity Manager, compare these files and copy the JARs manually to their destination directories. For an Oracle-shipped connector, details of the destination directories are given in the connector guide. See the Connector Code Files Changes section for more information.

5. Use the Upgrade Connectors feature.

   a. Log in to the Oracle Identity System Administration.

   b. In the left pane, under Provisioning Configuration, click **Manage Connector**.

   c. Use the Search feature to search for the source connector that you want to upgrade. In the table of search results, click the Upgrade icon for the source connector.

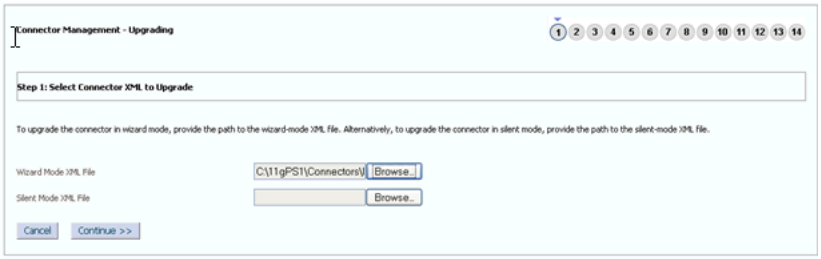
   d. On the Step 1: On the Upgrade page, select Connector XML for the Wizard Mode XML File field. Use the Browse option to navigate to the target version of the connector XML to which you want to upgrade.

      Make sure that you select the correct target connector XML. Upgrade feature does not validate the XML for target version or for any other connector object details. Leave Silent Mode XML File field empty. For example, if a user is upgrading the Active Directory connector from source version 9.1.1.7 to target version 11.1.1.5.0, user needs to select Active Directory 11.1.1.5.0 connector config XML (which is under xml folder) for Wizard mode upgrade XML field.

      Note: There is only one XML file for both trusted source reconciliation and target resource reconciliation for all the ICF based connectors. If you have more than one XML file, that is one for trusted source reconciliation and another for target resource reconciliation, you need to select the XML file for target resource reconciliation. Refer the connector guide (CI-XML) for the XML file name.

      Figure 11–25 shows the Select Connector XML to Upgrade page of the Connector Management - Upgrading wizard:

Figure 11–25 The Select Connector XML to Upgrade Page



e. Click **Continue**.

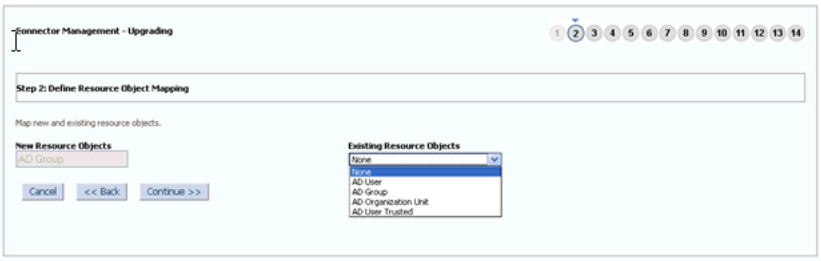
f. On the Step 2: Resource Object Mapping page, apply the following guidelines to map each new resource object with an existing resource object. Click Continue after you create each mapping.

- The New Resource Object field shows the name of a resource object in the target release. From the Existing Resource Object list, select the resource object in the source release to which you want to map the resource object in the target release. There might be a change in resource object names. It is your responsibility to map the resource object properly.
- If there are new resource objects that do not have a corresponding resource object in the source release, then select None from the Existing Resource Object list. This will happen only when the target connector versions add new resource objects that are not there in the source version.

Note: If you are upgrading from an Oracle-released source connector to an Oracle-released target connector, then see the connector guide for information about the mappings that you must create.

Figure 11–26 shows the Resource Object Mapping page of the Connector Management - Upgrading wizard:

Figure 11–26 The Resource Object Mapping Page



g. On the Step 3: Define Resource Scope page, a summary of the resource object mappings that you create is displayed. If there are resource objects in the source release that do not have corresponding resource objects in the target release, then they are displayed in the second table on this page. If you want to delete these resource objects, then select their check boxes. If a resource object is selected for deletion, then the resource will not be deleted from Oracle Identity Manager database. It just updates the OBJ_IS_SOFT_DELETE flag for the corresponding Resource Object to "1". The resource is still available for all provisioning and reconciliation. This flag is used in future.

Figure 11–27 shows the Define Resource Scope page of the Connector Management - Upgrading wizard:

Figure 11–27 The Define Resource Scope Page

This figure shows the Define Resource Scope page of the Connector Management - Upgrading wizard.

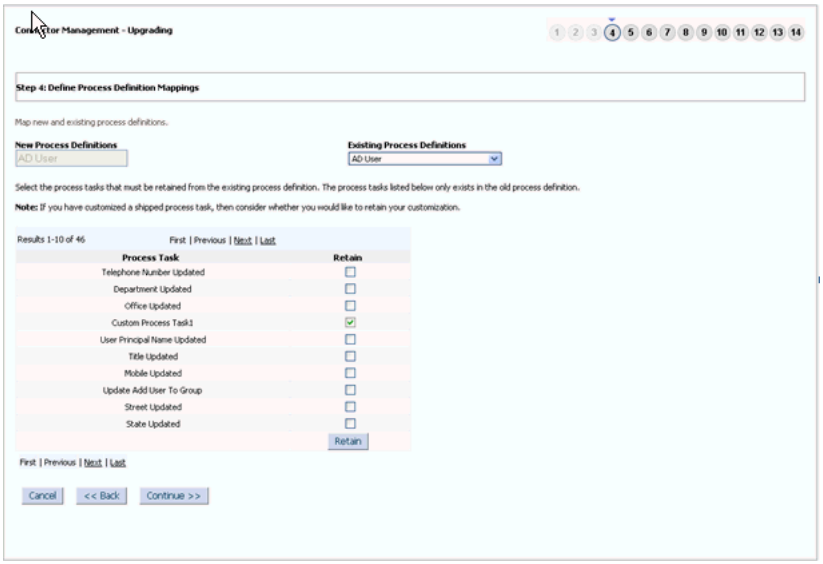
****

h. Click **Continue**.

i. On the Step 4: Define Process Definition Mapping page, map each new process definition with an existing process definition. Follow the guidelines given in Step f for mapping resource objects. Click Continue after you create each process definition mapping. If there are changes in the process definition names in source and target, it is your responsibility to map them properly. After selecting the corresponding source process definition for a specified target process definition, the page displays the list of process tasks available in the source process definition. You can retain the process tasks from the Source process definition. If there are any custom process tasks added to the source process definition, they can be retained. If there are any customization on the default process task, then before retaining such tasks you need to make sure there are no changes for this process task in the new connector release version by refereeing the connector guide. If a specific default process task is selected to retain, you might lose the changes (if there are any) for this process task in the new connector release. If the process tasks are part of the source connector and are not required in the target connector, then such process tasks must not be retained. It is recommended only to retain tasks that are added by user as part of customization of the source connector.

Figure 11–28 shows the Define Process Definition Mapping page of the Connector Management - Upgrading wizard:

Figure 11–28 The Define Process Definition Mapping Page



j. On the Step 5: Process Definition Mapping Summary page, a summary of the process definition mappings that you create is displayed. Click Continue to proceed.

Figure 11–29 shows the Process Definition Mapping Summary page of the Connector Management - Upgrading wizard:

Figure 11–29 The Process Definition Mapping Summary Page

k. On the Step 6: Define Form Mappings page, map each new form with an existing form. Follow the guidelines given in Step f for mappings resource objects. In addition, apply the following guideline and then click Continue after you create a mapping for each form. When a source process form is selected for each target, the page displays list of process form fields from the source process form attributes, which are not available in the target process form. These attributes either added to the source process as a part of customization or these were default attributes part of the source process form which may not be required for the target. You can select the attributes which are added as a part of customization, but need to verify if a default attribute is required in the target before retaining it.

Figure 11–30 shows the Define Form Mappings page of the Connector Management - Upgrading wizard:

Figure 11–30 The Define Form Mappings Page

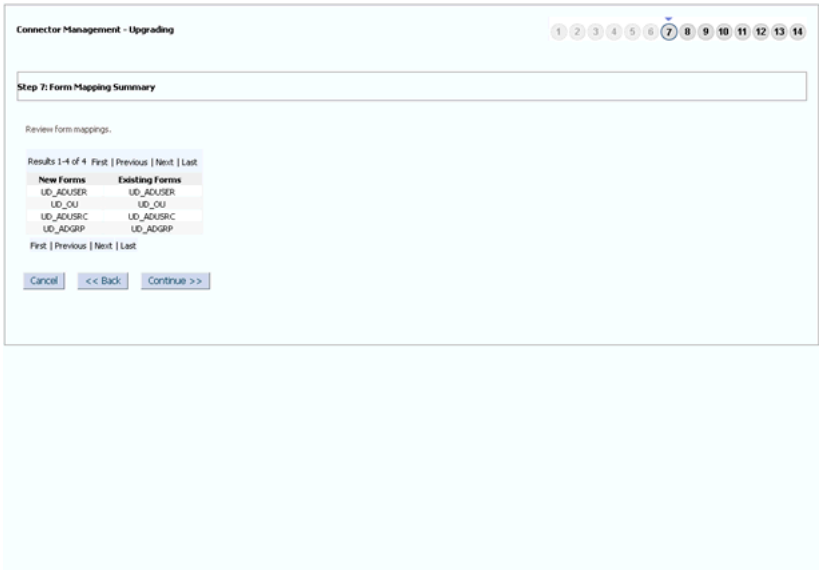
This figure shows the Define Form Mappings page of the Connector Management - Upgrading wizard.

****

l. On the Step 7: Form Mapping Summary page, a summary of the form mappings that you create is displayed. Click Continue to proceed.

Figure 11–31 shows the Form Mapping Summary page of the Connector Management - Upgrading wizard:

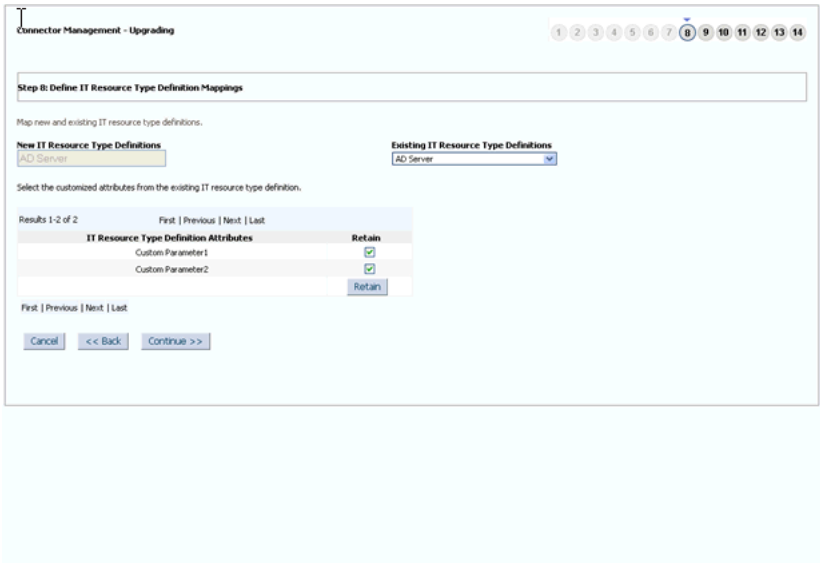
Figure 11–31 The Form Mapping Summary Page



m. On the Step 8: Define IT Resource Type Definition Mappings page, map each new IT resource definition with an existing IT resource definition. Follow the guidelines given in Step f for mappings resource objects. Click Continue after you create a mapping for each IT resource definition. If there are changes in the names of the IT resource type definition, then it is your responsibility to map them properly. Refer the connector guide to check the change in default IT resource type definition names. When a target IT resource type definition is mapped with corresponding source IT resource type definition, the page displays list of IT resource type definition parameters, which are part of source definition but not available in target definition. These are either added as a part of customization or they were part of source definition. If these parameters are added as part of customization, then you need to retain them.

Figure 11–32 shows the Define IT Resource Type Definition Mappings page of the Connector Management - Upgrading wizard:

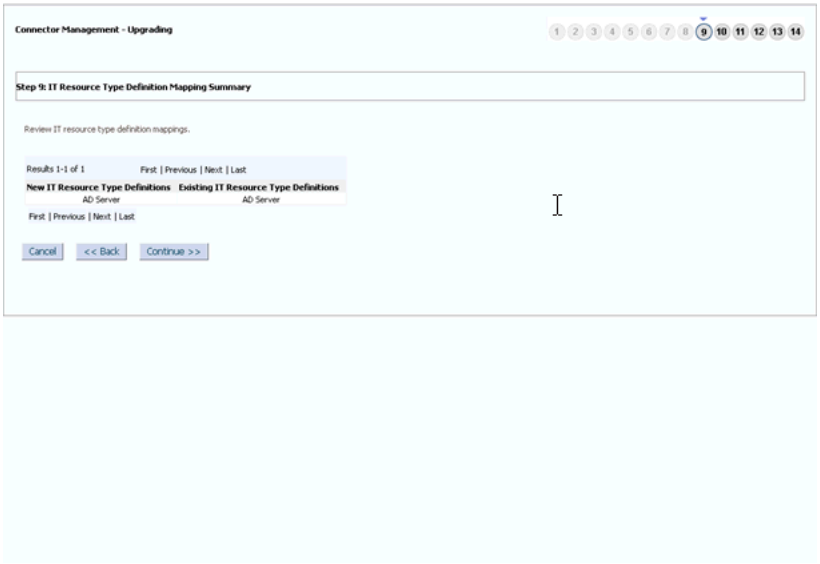
Figure 11–32 The Define IT Resource Type Definition Mappings Page



n. On the Step 9: IT Resource Type Definition Mapping Summary page, a summary of the IT resource type definition mappings that you create is displayed. Click Continue to proceed.

Figure 11–33 shows the IT Resource Type Definition Mapping Summary page of the Connector Management - Upgrading wizard:

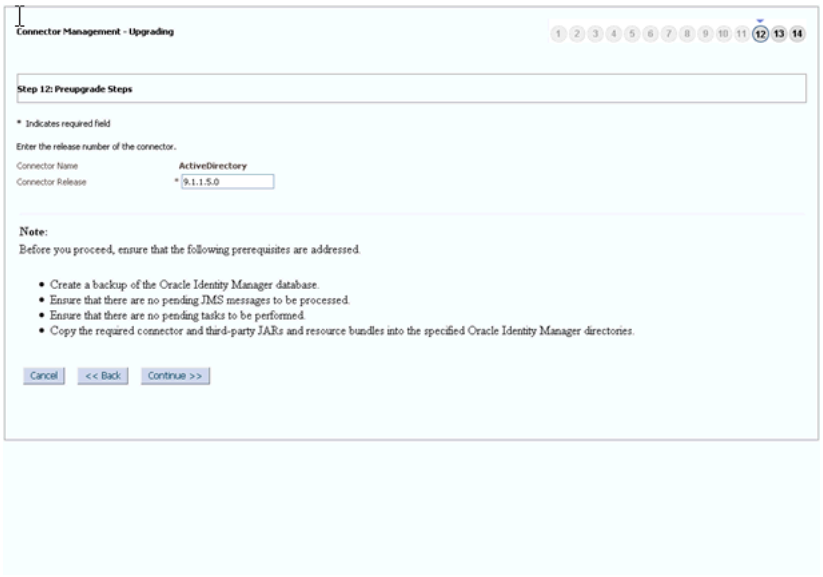
Figure 11–33 The IT Resource Type Definition Mapping Summary Page



o. On the Step 12: Preupgrade Steps page, enter a new release number for the connector in the Connector Version field. Click Continue to proceed. The upgrade process does not validate the version provided with the connector release version. You need to provide correct version here by referring the connector guide.

Figure 11–34 shows the Preupgrade Steps page of the Connector Management - Upgrading wizard:

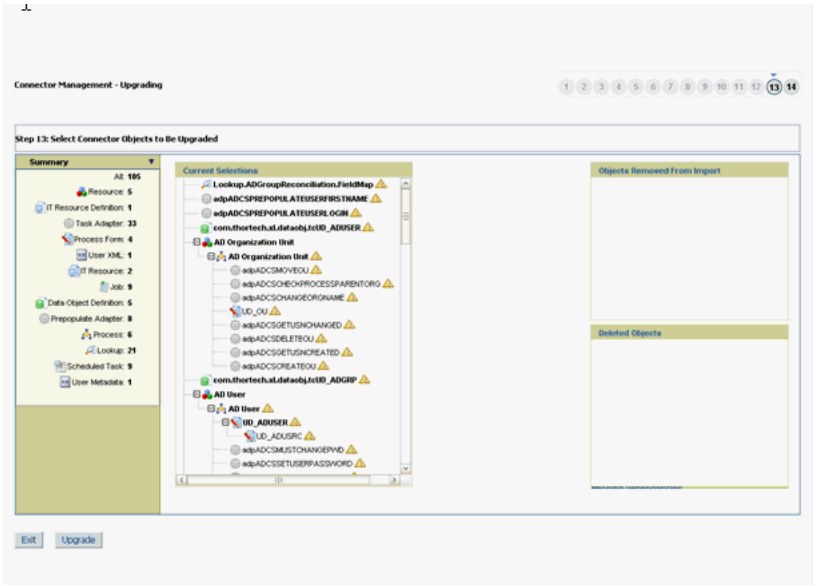
Figure 11–34 The Preupgrade Steps Page



p. On the Step 13: Select Connector Objects to Be Upgraded page.

Figure 11–35 shows the Select Connector Objects to Be Upgraded page of the Connector Management - Upgrading wizard:

Figure 11–35 The Select Connector Objects to Be Upgraded Page

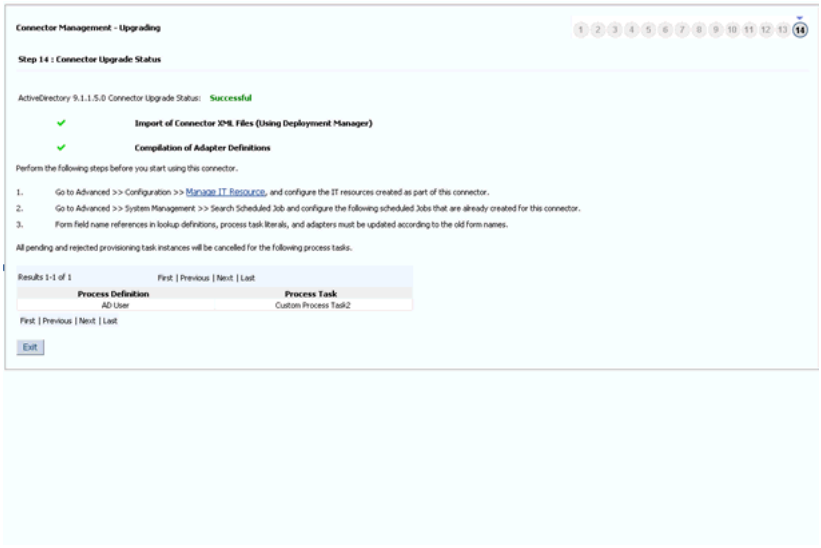


Note: If the Connector Management - Upgrading wizard is opened by using Microsoft Internet Explorer, then all the fields and buttons on the Step 13: Select Connector Objects to Be Upgraded page might not be visible. There is no scroll bar available in the page. Therefore, maximize the window to display all the controls in the page.

q. After you review the information on the Connector Upgrade Status page, click **Upgrade** to start the upgrade process.

Figure 11–36 shows the Connector Upgrade Status page of the Connector Management - Upgrading wizard:

Figure 11–36 The Connector Upgrade Status Page



Note down the process definition names and the corresponding process task names. These process tasks are not going to be used by Oracle Identity Manager anymore. Therefore, all their pending and rejected instances need to be canceled.

Use cancelProcessTask utility available in *OIM_HOME*/server/bin. The utility takes the process definition name and the process task name as input. You need to run the utility for each process task.

The Upgrade Connectors feature processes connector object mappings in the following manner:

- If a new connector object is mapped to None, then the new connector object is inserted in the database.
- A new resource object, process definition, or form replaces the old resource object, process definition, or form to which it is mapped.
- The new names of the process form are converted into the old process form names.
- If an old and a new lookup definition have the same name, then their contents are merged.
- When the Upgrade Connectors feature tries to delete an object, which is not going to be used by upgraded version of connector, an exception is thrown if the instances of the object exists in Oracle Identity Manager database. Such an object is renamed and soft deleted so that it will not be used anymore by Oracle Identity Manager.

6. Perform the following steps:

a. Change form names and form field column name references in the following objects:

Note: For an Oracle-released connector, see the connector guide for information about the changes to be made.

- Lookup definitions
- Process task literals
- Adapter literals

b. All the default adapters are overwritten. Therefore, if customer has done any customization, the changes need to be applied after connector upgrade.

c. After the upgrade, contents of existing and new lookup definitions are merged. In these lookup definitions, you must manually delete entries that are not required.

7. Verify that all use cases specific to the target are working fine including provisioning and reconciliation.
8. Generate the XML file. This XML file contains details of the object definition changes from the source release to the target release.

   To generate this file:

   a. Log in to the Oracle Identity System Administration.

   b. In the left pane, under Provisioning Configuration, click **Manage Connector**.

   c. Use the Search feature to search for the connector.

   d. In the search results table, click the Export Silent Upgrade XML icon for the connector.

   e. Specify the location where you want the file to be saved.

   Note: If the upgrade fails, then perform the following steps:

   1. Look at the exception and take suitable action.
   2. Restore the Oracle Identity Manager database and MDS.
   3. Proceed for the upgrade.

Silent Mode Upgrade in Staging and Production Environment

Note: You need to perform preupgrade and post upgrade steps while performing silent mode upgrade.

Caution: Before you import the XML file, verify that the source connector on the production server is the same as the source connector on the staging server. If there are differences in the source connector on the staging server and the production server, then the XML file is not correctly imported on the production server.

To perform the silent mode upgrade on the production server:

1. Copy the XML file to the host computer of the Oracle Identity Manager installation on which you want to import the file. Alternatively, copy the XML file

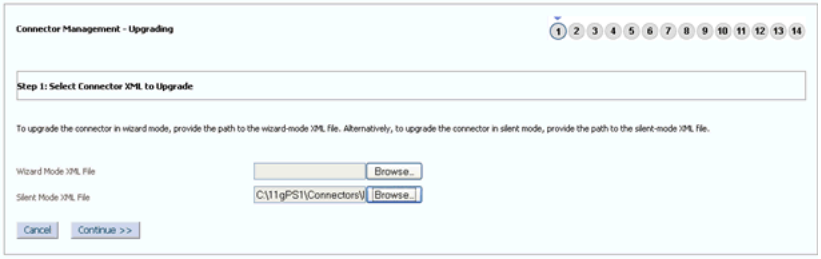
to a shared folder on another computer that can be accessed from the Oracle Identity Manager host computer.

2. Log in to the Oracle Identity System Administration.
3. In the left pane, under Provisioning Configuration, click **Manage Connector**.
4. Use the Search feature to search for the source connector that you want to upgrade.
5. In the table of search results, click the Upgrade icon for the source connector.
6. On the Step 1: Select Connector XML to Upgrade page of the utility, click Browse and navigate to the connector XML file for the source release in the silent mode upgrade XML field.

> **Note:** There is only one XML file for both trusted source reconciliation and target resource reconciliation for all the ICF based connectors. If you have more than one XML file, that is one for trusted source reconciliation and another for target resource reconciliation, you need to select the XML file for target resource reconciliation. Refer the connector guide (CI-XML) for the XML file name.

Figure 11–37 shows the Select Connector XML to Upgrade page of the Connector Management - Upgrading wizard:

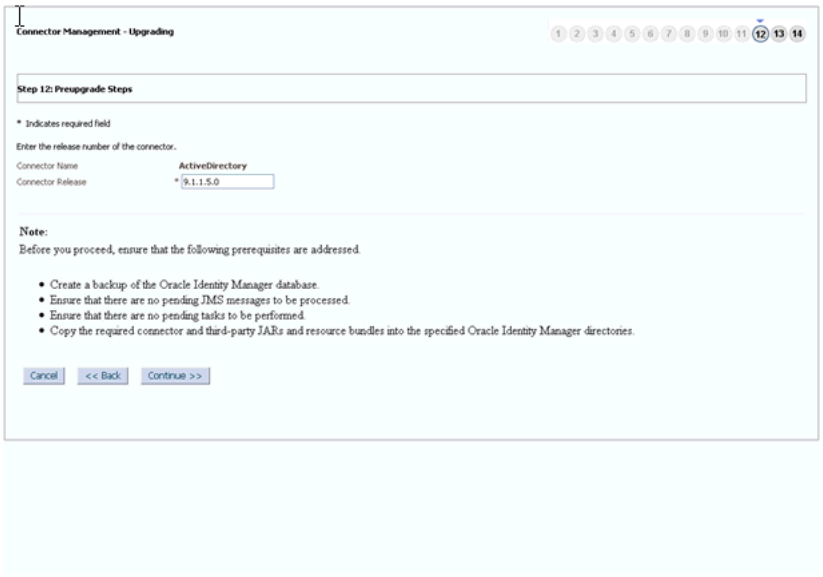
Figure 11–37 The Select Connector XML to Upgrade Page



7. Click **Continue**.
8. On the Step 12: Preupgrade Steps page, click **Continue** to proceed.

   Figure 11–38 shows the Preupgrade Steps page of the Connector Management - Upgrading wizard:

Figure 11–38 The Preupgrade Steps Page



9. On the Step 13: Select the Connector Objects to be Upgraded page, review the summary of the connector objects that you selected for upgrade.

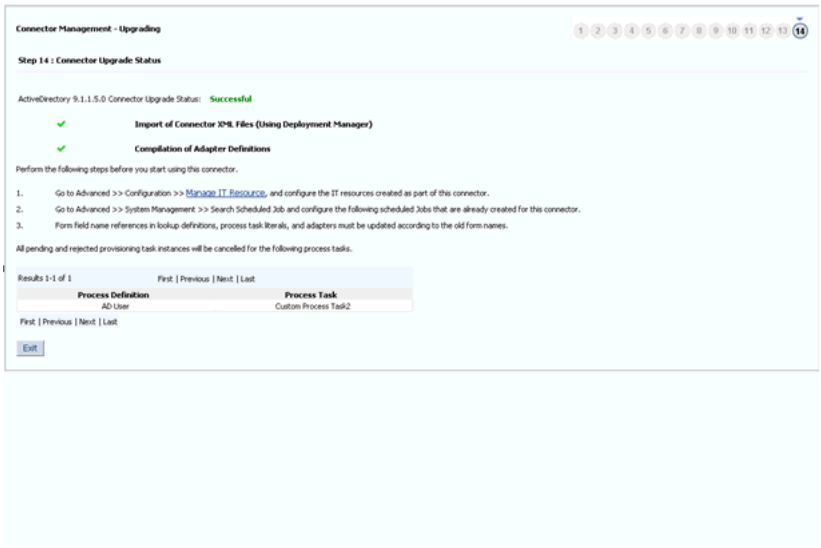
   Figure 11–39 shows the Select the Connector Objects to be Upgraded page of the Connector Management - Upgrading wizard:

Figure 11–39 The Select the Connector Objects to be Upgraded Page

10. After you review the information on the page, click **Upgrade** to start the upgrade process.

    The Connector Upgrade Status page shows the status at the end of a successful upgrade, as shown in Figure 11–40:

Figure 11–40 The Connector Upgrade Status Page



11.8.5.3 Postupgrade Procedure

The following sections describe procedures that you must perform after the upgrade operation:

- Connector Code Files Changes
- Running the PurgeCache Utility
- Running cancelProcessTask Utility
- Updating Access Policies
- Configuring the IT Resource
- Configuring the Scheduled Tasks
- Other Postupgrade Steps

Connector Code Files Changes

During an upgrade operation, you need copy connector code files, which include JAR files and scripts to the specified directories. To do so:

1. Manually upload all the connector specific jars (excluding common library files Common.jar, FAMILYCommon.jar, and icf-Common.jar) present in the "lib" folder of the connector distribution bundle using UpdateJars utility (available under *OIM_HOME*/server/bin) to Oracle Identity Manager database. Before running the UpdateJars utility, set *APP_SERVER*, *OIM_ORACLE_HOME*, *JAVA_HOME*, *MW_HOME*, *WL_HOME*, and *DOMAIN_HOME*.
2. Download common library (Common.jar, FAMILYCommon.jar and icf-Common.jar) from Oracle Identity Manager database using DownloadJar utility (available under *OIM_HOME*/server/bin).
3. Extract MANIFEST.MF from the downloaded libraries. Compare this version of MANIFEST.MF with the version in MANIFEST.MF of the common libraries that is available as part of ICF based distribution bundle. If the distributed library version is higher than the one downloaded from Oracle Identity Manager database, then use the UploadJar utility (available under *OIM_HOME*/server/bin) to upload the common libraries to Oracle Identity Manager database.

Running the PurgeCache Utility

When the upgrade is performed, there might be stale data in the cache, which is required to be purged. The PurgeCache utility purges the cache. See *Oracle Fusion Middleware Performance and Tuning Guide* for information about purging the cache.

Note: Before running this utility, set *APP_SERVER*, *OIM_ORACLE_HOME*, *JAVA_HOME*, *MW_HOME*, *WL_HOME*, and *DOMAIN_HOME*.

Running cancelProcessTask Utility

This utility is used for canceling the pending and rejected instances of a process task. If a process task of a process definition, which is there in the source connector and is not required in the target, then the process task is soft deleted in the upgrade process. Oracle Identity Manager will not use such soft deleted task as part of provisioning work flow after upgrade. All the instances of such deleted process task, which are in pending and rejected status need to be canceled.

The utility is available in *OIM_HOME*/server/bin. This utility will take the process task name and the corresponding process definition name as input.

Note: Before running this utility, set *APP_SERVER*, *OIM_ORACLE_HOME*, *JAVA_HOME*, *MW_HOME*, *WL_HOME*, and *DOMAIN_HOME*.

Updating Access Policies

In Oracle Identity Manager, an access policy is associated with a resource object. While creating an access policy, user would have provided the data for the process form attributes. As the part of connector upgrade, if there are changes in the form attributes, then you need to edit the access policy to check the data for the existing and the new fields. For example, if the connector upgrade adds a new process form attribute, you can provide the data for the new attribute by editing the access policy.

Configuring the IT Resource

Verify that the IT resource instances have proper values after upgrade.

Configuring the Scheduled Tasks

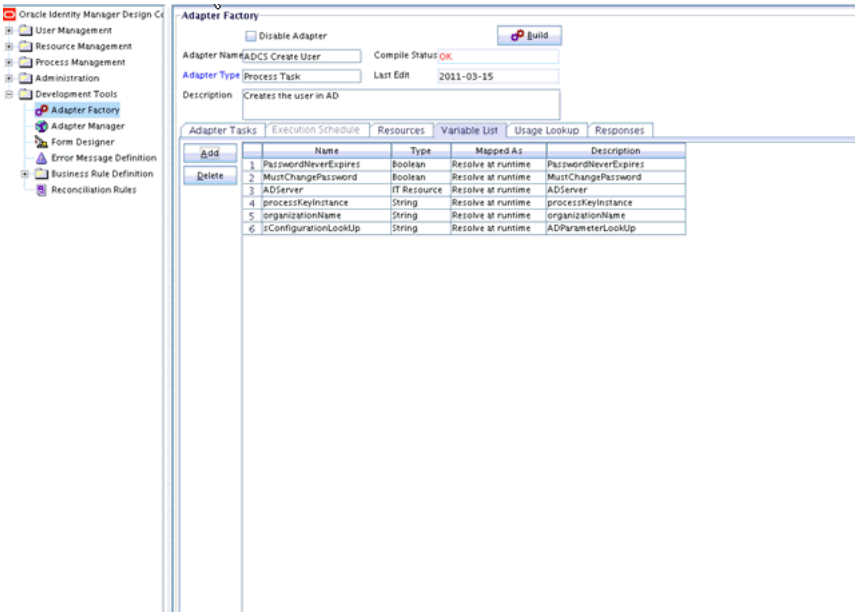
Set values for attributes of the scheduled tasks of the target release. For an Oracle-released target connector, see the connector guide for information about the scheduled task attributes.

Update Adapters for Changes in IT Resource Type Definition Parameter

If there are changes in the IT Resource Type Definition Parameter names, you need to update the custom adapters for the parameter changes. To do so:

1. Log in to Design Console.
2. Open the custom adapter using the adapter factory.
3. Go to the variable list and check if there are any variables of type IT Resource, as shown in Figure 11–41:

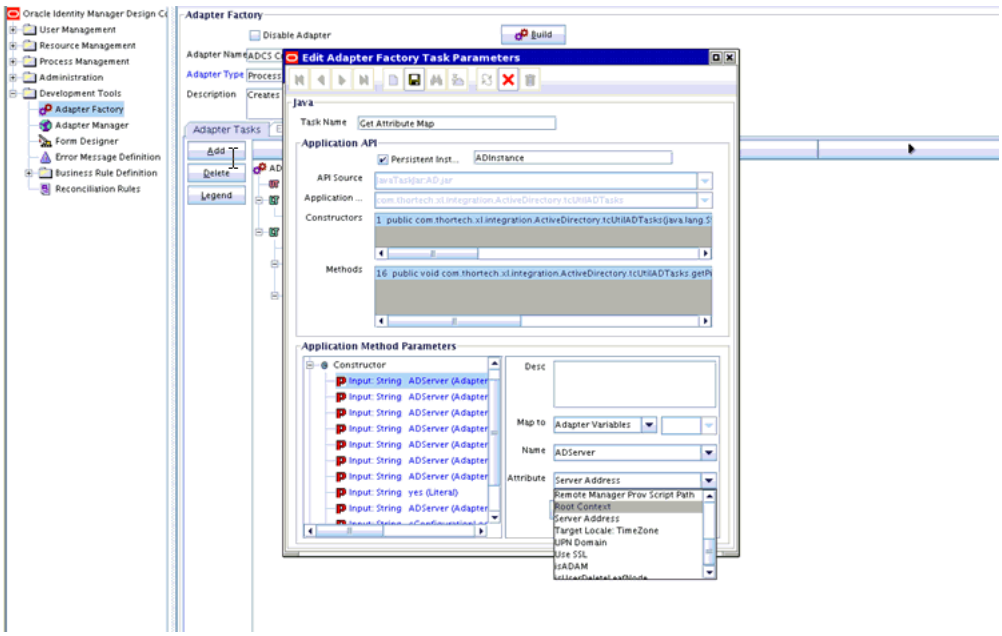
Figure 11–41 The Variable List Tab of the Adapter Factory Form



4. If there is a variable of IT Resource, then go to the task details and change the mapping of the IT Resource parameter mapping to the new target field (if the parameter is changed/deleted).

   Figure 11–42 shows the Edit Adapter Factory Task Parameters dialog box that enables you to change the mapping of the IT Resource parameter mapping to the new target field:

Figure 11–42 The Edit Adapter Factory Task Parameters Dialog Box



5. If the adapter is mapped to the IT Resource Type Definition parameter, then you need to verify if the mapped parameter is not deleted. If the parameter is deleted, then you need to remap it to the correct parameter.

To verify the adapter mappings:

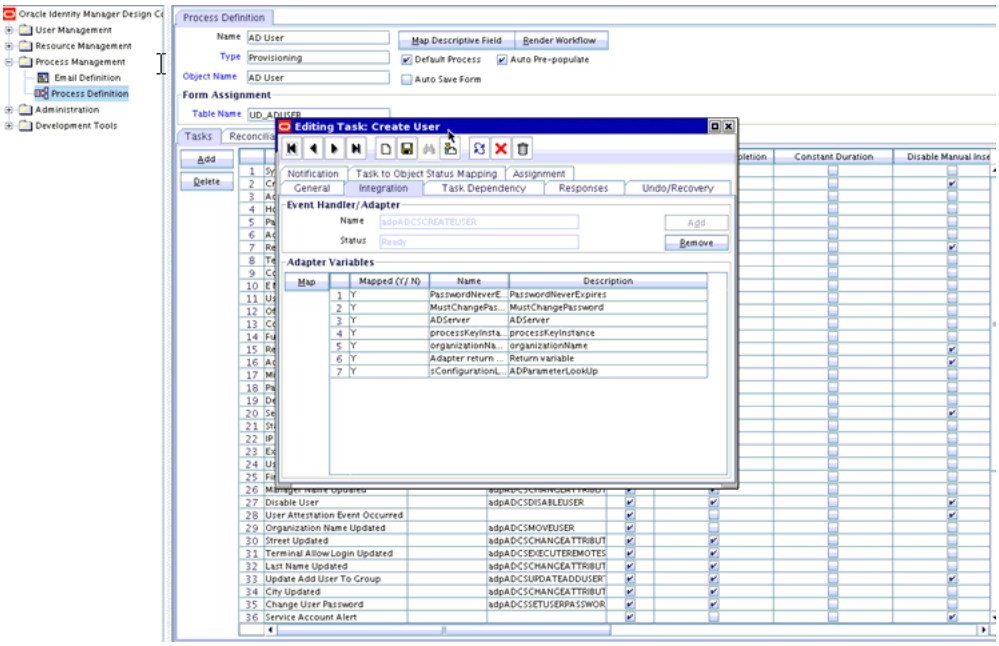
a. Verify the mapping for process task adapter as follows:

i) Log in to Design Console.

ii) Go to Process Definition.

iii) Click the task, and then click the **Integration** tab, as shown in Figure 11–43:

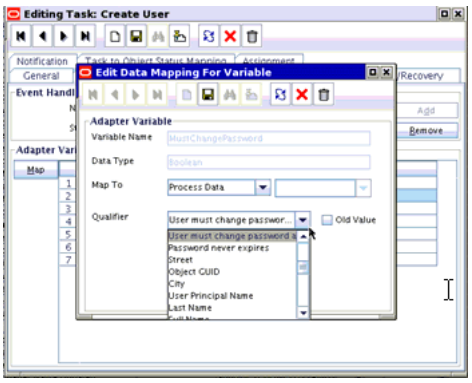
Figure 11–43 The Integration Tab of the Editing Task Dialog Box



iv) Check if the adapter variable is mapped to the deleted/modified form attribute. If yes, remap such attributes to adapter variables. Repeat this step for all process tasks of all process definitions of the connector.

Figure 11–44 shows the Editing Data Mapping for Variable dialog box that enables you to view and edit the adapter variable mapping to the form attribute:

Figure 11–44 The Editing Data Mapping for Variable Dialog Box

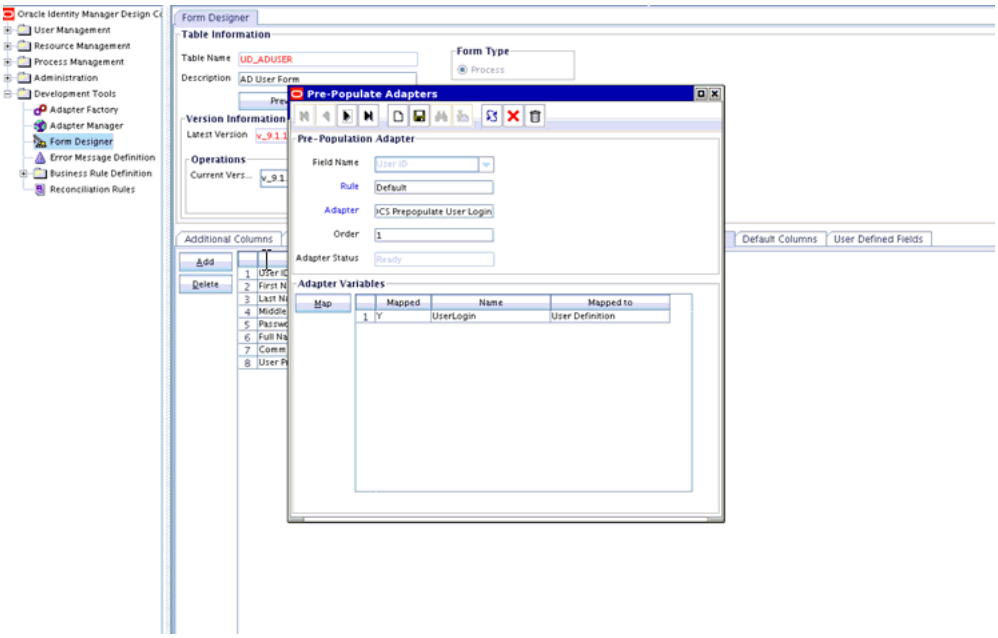


b. Prepopulate adapter mappings as follows:

i) Log in to Design Console.

ii) Go to Form Designer, Pre-Populate Adapters, as shown in Figure 11–45:

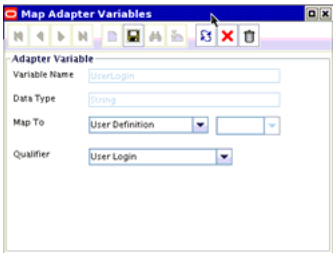
Figure 11–45 The Pre-Populate Adapters Dialog Box



iii) Click **Map** to map adapter variable and check if any of the fields are mapped to the process data attributes. If it is mapped, then verify the process form attribute is not deleted as part of upgrade. If the process form attributes are deleted, then remap them to the correct form attribute data.

Figure 11–46 shows the Map Adapter Variable dialog box:

Figure 11–46 The Map Adapter Variable Dialog Box



Note: Repeat the procedure for all the prepopulated fields of all the process forms of the connector. If there are any entity adapter, then check the adapter variables mapping for these adapters in Data Object Manager.

Other Postupgrade Steps

Perform the following postupgrade steps:

1. Change form names and form field column name references in the following objects:

Note: For an Oracle-released connector, see the connector guide for information about the changes to be made.

- Lookup definitions
- Process task literals
- Adapter literals

2. Verify all the reconciliation fields on the resource object and corresponding reconciliation form field mapping on the process definition. Delete old default reconciliation fields, if there are any, which have mapping to the process form fields that are not retained as part of upgrade.
3. Verify that upgrade process has retained all customizations, for example, customizations on Resource Object, Process definition, and Process Form.
4. After the upgrade, contents of existing and new lookup definitions are merged. In these lookup definitions, you must manually delete entries that are not required.
5. Run the Lookup reconciliation again. The old lookup reconciliation data is available in the Lookups after upgrade. Re-running the Lookups is required if there is a change in the format for the lookup values. Refer the specific connector guide for more details about lookup reconciliation.
6. Recalculate statistics and re-create indexes and other database objects that are removed or made invalid by the upgrade process. For more information, see Oracle Identity Manager Database guide.
7. Check adapters status related to the connectors. If the adapters are not compiled, then you must compile them.
8. Verify that the custom parameters are available after upgrade. Custom Scheduled Task parameters are retained as part of upgrade process. Modify the scheduled task to add the parameter if it is not available after upgrade.
9. Verify if there are any changes in the application forms. If yes, then delete the existing forms for the resource. Modify the new application forms for any customization.

11.8.6 Procedure to Upgrade a 9.x Connector Version to an ICF Based Connector

ICF based Connector provides LCM as a new feature that uses Connector Installer to import the connector, where as 9.*x* connector uses Deployment Manager to import definitions of the objects that constitute a connector. Since LCM offers a more broader and richer feature in installing and/or managing a connector than Deployment Manager, it is recommended to use only Connector installer for Oracle Identity Manager 11*g* connectors installation and/or management.

To upgrade a 9x connector version to a ICF based connector:

1. Delete all the existing jar files such as Javatasks, ScheduleTask, and ThirdParty jars related to the 9x connector except for the Common.jar file.
2. Download Common.jar and extract its MANIFEST.MF. Compare this version of MANIFEST.MF with the version in MANIFEST.MF of the Common.jar that is available as part of ICF based connectors distribution bundle. Retain/Upload (using UploadJars utility) Common.jar in Oracle Identity Manager database that has higher version.

3. Manually upload all the jars present in the "lib" folder of the ICF based connector distribution bundle using the UploadJars utility in Oracle Identity Manager database (available under *OIM_HOME*/server/bin).
4. Explode the connector bundle (with naming convention "org.identityconnectors.*") in some temporary folder. Make a folder named "lib" in the same temporary folder and copy all the third party libraries to that folder.
5. Retain MANIFEST.MF from the above exploded bundle.
6. Repackage the connector with the same name and with the same MANIFEST.MF that was being retained. Now, the repackaged connector bundle will also be having third party libraries.
7. Upload the repackaged connector in Oracle Identity Manager database with jar type as "ICFBundle".
8. Delete the temporary folder created in Step 4.
9. Upgrade the connector by following the upgrade process
10. Purge cache or restart the server.

11.9 Uninstalling Connectors

Connector uninstall utility deletes the data related to the connector chosen for uninstall from Oracle Identity Manager Database. It deletes all the account related data associated with resource objects of the connector.

This utility does not delete:

- The actual user account from the target system
- Identities from Oracle Identity Manager although the users are brought from trusted source to Oracle Identity Manager through trusted reconciliation
- Audit data
- Archival data

Connector uninstall utility does not validate and notify the user if there is any object dependency present. For example, while uninstalling a Microsoft Active Directory (AD) connector, it does not validate if a dependent connector, such as Microsoft Exchange connector, already exists or not. Before uninstalling a connector, you must check if there are any other connectors dependent on the connector. If there are any, then the connector must not be uninstalled because this will affect the functionality of the dependent connectors. You must uninstall all the dependent connectors before uninstalling the base connector.

This section discusses the following topics:

- Use Cases Supported by the Uninstall Connectors Utility
- Overview of the Connector Uninstall Process
- Setting Up the Uninstall Connector Utility
- Uninstalling Connectors and Removing Connector Objects

11.9.1 Use Cases Supported by the Uninstall Connectors Utility

The following use cases are supported by the Uninstall Connectors utility:

- A target system that has been decommissioned, and you want to uninstall the connector that was used to link that target system with Oracle Identity Manager.
- Instead of directly upgrading to the latest release of a connector, you want to uninstall the earlier release and then perform a fresh installation of the latest release.
- You want to remove an individual connector object from the Oracle Identity Manager database. For example, you had created a resource object in Oracle Identity Manager to represent the Intern user type defined in your target system. This user type has been removed from the target system, and you now want to remove the resource object from Oracle Identity Manager.

  The Uninstall Connectors utility supports independent deletion of following connector artifacts:

  – Adapters
  – Lookup definitions
  – Resource objects
  – Scheduled tasks

11.9.2 Overview of the Connector Uninstall Process

When you run the Uninstall Connectors utility, the utility performs the following steps before deleting the resource objects of the connector:

1. Checks if there are any access policies associated with the resource objects of the connector. If there are any access policies present, then the utility displays the list of access policies associated with the resource object and prompts you to modify the access policy and terminates with no data deletion. The access policy should be modified to remove the resource object from it. If the access policy is associated with only one resource object, then you need to create a dummy resource object, assign it to the access policy and then proceed with the removal of resource object from the access policy.
2. Closes all requests associated with the resource objects.
3. Displays the list of attestation processes which are associated with the resource objects. Attestation processes are generic in nature, therefore the utility does not delete attestation processes from Oracle Identity Manager. It prompts you to modify these processes as the resource objects would be deleted from Oracle Identity Manager.

The following objects that constitute the connector are dropped from the Oracle Identity Manager database.

1. Resource object and objects related to the resource object.
   a. Entitlement assignment, entitlement assignment history, and entitlement data
   b. Tasks and task history associated with any provisioning process linked to the resource object
   c. Process forms associated with the resource object
   d. Process instance and object instances associated with the resource object
   e. Reconciliation events and data associated with the resource object
   f. Attestation event data for the resource object
   g. Requests and request data associated with the resource object

h. E-mail definitions for the resource object
i. Entitlements associated with the resource object
j. Regular rules associated with the resource object
k. Reconciliation owner matching rules for the resource object
l. Reconciliation action rules for the resource object
m. Status codes corresponding to this resource object
n. Reconciliation process mappings for the resource object
o. Reconciliation object fields for the resource object
p. Application form to process form mappings for the resource object.
q. Object dependency tables for parent and child forms for the resource object
r. Resource object for organization
s. Process determination rules associated with the resource object
t. Password policy rules associated with the resource object
u. IT resource instances that are associated with IT resource types defined on forms that are linked to provisioning processes. If there is any default IT resource instance, they will not be deleted, for example, IT resource instance of Remote Manager
v. Process instances and resource object instances
w. Tasks associated with the provisioning processes
x. The actual object and process, parent and child tables associated with the resource object.

2. Scheduled tasks and scheduled jobs
3. Adapters/Event Handlers
4. Lookup definitions

11.9.3 Setting Up the Uninstall Connector Utility

To set up the Uninstall Connector utility:

- Files that constitute the Uninstall Connector utility are viable in *OIM_HOME*/server/bin directory. These files are as follows:
  - ConnectorUninstall.properties
  - uninstallConnector.bat
  - uninstallConnector.sh

11.9.4 Uninstalling Connectors and Removing Connector Objects

Depending on your requirements, you can use the Uninstall Connectors utility to perform any of the following tasks:

- Uninstalling a Connector
- Removing Adapters, Lookup Definitions, Resource Objects, and Scheduled Tasks

The following section provides detailed instructions on using the ConnectorUninstall script to delete connector objects from the Oracle Identity Manager database. Each of the earlier sections provides a link to this section.

- Running the Script to Uninstall Connectors and Connector Objects

11.9.4.1 Uninstalling a Connector

Caution: It is strongly recommended that Oracle Identity Manager is idle and it is not available for any operations. You must ensure that:

- There are no operations on Oracle Identity Manager while using uninstalling connector or connector objects
- All scheduled tasks are disabled and there are no asynchronous messages pending for processing such as audit messages, offline provisioning messages, offline task messages, requests scheduled for future and so on.

You can use the ConnectorUninstall script to uninstall a connector. When you run the script, all objects that form part of the connector and all the resource data that was collected through the connector are deleted from the database.

Note: Before running the uninstall utility:

- You cannot delete data that are already archived.
- You must ensure that you have the latest Oracle Identity Manager schema and MDS backup, which will help to restore if uninstall utility does not complete successfully.
- You must ensure that your UNDO tablespace is sized properly. This is required if your development/test environment has significant amount of data to be deleted.

As mentioned earlier in this guide, when a connector is defined, an entry is created for the connector in the Oracle Identity Manager database. This entry also includes the contents of the connector XML. When you choose to uninstall a connector, the utility identifies the connectors objects to be dropped by parsing the connector XML contents.

Warning:

- Connector uninstall collects all the objects information from the connector XML, which is created while installing or defining a connector. If an additional object, which is not related to this connector is added while defining the connector, uninstall would delete that too. For example, while defining AD connector, if user adds a system lookup or lookup related to other connector, uninstall would delete that lookup.
- Ensure that only the connector specific objects are added while defining a connector.

See "Running the Script to Uninstall Connectors and Connector Objects" on page 11-67 for the procedure.

11.9.4.2 Removing Adapters, Lookup Definitions, Resource Objects, and Scheduled Tasks

Caution: It is strongly recommended that Oracle Identity Manager is idle and it is not available for any operations. You must ensure that:

- there are no operations on Oracle Identity Manager while using uninstalling connector or connector objects
- all scheduled tasks are disabled and there are no asynchronous messages pending for processing such as audit messages, offline provisioning messages, offline task messages, requests scheduled for future and so on.

You can use the ConnectorUninstall script to remove an adapter, lookup definition, resource object, or scheduled task. Only the object that you specify is removed from Oracle Identity Manager.

11.9.4.3 Running the Script to Uninstall Connectors and Connector Objects

Running the script to uninstall connectors and connector objects includes the following procedures:

- Preuninstall
- Uninstall
- Postuninstall

11.9.4.3.1 Preuninstall

Note: Before executing the uninstall, you must ensure that all scheduled tasks are disabled.

Before Uninstalling the connector, you must:

1. Create a backup of Oracle Identity Manager database so that if something goes wrong during uninstalling, then the data can be restored. See Oracle Identity Manager Database documentation for details about creating database backup.
2. Create Oracle Identity Manager metadata (MDS) backup.
3. Ensure that there are no operations on Oracle Identity Manager until the Uninstall utility is completed. Oracle Identity Manager and SOA servers should be up and running.
4. Ensure that all the JMS messages are processed.

11.9.4.3.2 Uninstall To run the ConnectorUninstall script for uninstalling the connector:

1. Set values in the properties file used by the script.

Note: If you provide ConnectorName and Release along with ObjectType and ObjectValues, then deletion of ObjectValues is performed by the utility and the Connector information is skipped.

The ConnectorUninstall.properties file is a viable in *OIM_HOME*/server/bin. This file contains information that is used by the script for deleting connector objects.

Open the properties file in a text editor, and then set values for the following properties:

- DatabaseURL: Enter the JDBC URL for the Oracle Identity Manager database in the following format:

  ```
  jdbc:oracle:thin:@HOST_NAME:DATABASE_PORT:DATABASE_NAME/ORACLE_SID

  For example: jdbc:oracle:thin:@localhost:1521:orcl
  ```

- DBUserName: Enter the user name of an Oracle Identity Manager database.
- DBType: Specifies the type of database.
- LogLevel: Enter one of the following as the log level: DEBUG, WARN, INFO, or ERROR.
- Location: Enter the directory location where you want to have all the log files generated by the Uninstall utility.

  If the Uninstall utility completes successfully, then the ConnectorUninstall.log file, along with <ResourceObject>.log files are generated.

  If the Uninstall utility fails, then the ConnectorUninstall.log file along with the ConnectorUninstall_Error.log file are generated.

  Note: If the uninstall utility fails with errors, then check the ConnectorUninstall.log and ConnectorUninstall_Error.log and take suitable action. Then, run the uninstall utility again.

  For example, if the Uninstall utility of ActiveDirectory Connector succeeds, then the following logs are generated:

  – ConnectorUninstall.log
  – AD User.log
  – AD Group.log
  – AD Oraganization Unit.log
  – AD User Trusted.log

  If the Uninstall utility of ActiveDirectory Connector Fails, then the following logs are generated:

  – ConnectorUninstall.log
  – ConnectorUninstall_Error.log

- ConnectorName: The value that you set for this property depends on your requirement. If you want to delete a specific connector, then enter the name of the connector. The name that you enter must be the same as the name shown in the search results displayed through the Manage Connector feature. For example, enter `Active Directory` if you want to delete the Microsoft Active Directory connector.
- Release: The value that you set for this property depends on your requirement. If you want to delete a specific connector, then enter the release number of the connector. The release number that you enter must be the same

as the release number shown in the search results displayed through the Manage Connector feature. For example, enter `9.1.0.1` if you want to delete the Microsoft Active Directory 9.1.0.1 connector.

- ObjectType: The value that you set for this property depends on your requirement:
  - If you want to uninstall a connector, then ensure that the ObjectType property is not assigned a value.
  - If you want to delete adapters, lookup definitions, resource objects, or scheduled task, then enter `Adapter, Lookup, ResourceObject,` or `ScheduledTask` respectively.

    Example: ResourceObject
- ObjectValues: Enter a semicolon-separated list of object values.

  Example: AD User; AD Group

2. In a command window, change to the *OIM_HOME*/server/bin directory and then run the script, sh uninstallConnector.sh (or bat file).

   Note: Before running this utility, set *APP_SERVER, OIM_ORACLE_HOME, JAVA_HOME, MW_HOME, WL_HOME,* and *DOMAIN_HOME*.

   While the script runs, logs are generated at the location provided.

   After you run the utility, you are prompted to enter following information:

   a. Oracle Identity Manager Database Password
   b. Oracle Identity Manager Administrator Name
   c. Oracle Identity Manager Administrator Password
   d. Oracle Identity Manager Server t3 URL

      For example: t3://<*HOST_NAME*>:<*HOST_PORT*>

      Note: For cluster setup, the t3 URL should be t3://<*NODE1*>:<*PORT1*>,<*NODE2*>:<*PORT2*>.

   e. Context Factory
   f. Confirmation for the deletion of the connector/object(s)

11.9.4.3.3 Postuninstall After uninstalling the connector, you must perform the following steps:

1. Use DeleteJars utility for deleting the jars associated with the connector from Oracle Identity Manager database.
2. Use DeleteResourceBundles utility for deleting all resources that are associated with the connector from Oracle Identity Manager database.
3. Revisit the log, look for the following information and perform the steps mentioned for each of it:
   a. The list of attestation processes: Delete/modify these attestation process as the resource objects, which used these attestation processes are now deleted.

b. Modify requests manually to delete the resource object names that are cleaned by the uninstall utility.

c. As the part of connector uninstall, the approval processes (Approval workflow/SOA composites) are not deleted. If the approval processes are generic, then you need to modify them if they have association with the deleted resource objects.

4. Recalculate statistics and re-create indexes and other database objects that are removed by the connector uninstall utility. For more information, see "Performance Tuning and Best Practices".

5. Restart Oracle Identity Manager, or use PurgeCache utility to purge the Cache.

   See *Oracle Fusion Middleware Performance and Tuning Guide* for information about purging the cache.

11.10 Troubleshooting Connector Management Issues

Problem

Using Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0), you can configure a cloned Active Directory (AD) Release 9.x connector for target AD and run an AD trusted source reconciliation to create users in Oracle Identity Manager. After the user is created in Oracle Identity Manager, when you run the target resource reconciliation for AD, the user details are linked in the Accounts tab. However the Detail Information tab displays a blank page. When you check the Application Instances section in Oracle Identity System Administration and search and open the relevant application instance, no form is found associated with the application instance.

Solution

Create a new set of forms for each application instance.

Problem

When you are upgrading a connector, the following error may be encountered by Oracle Identity Manager:

```
<Error> <XELLERATE.WEBAPP> <BEA-000000> <Class/Method:tcActionBase/execute
encounter some problems: Bean has been deleted.
javax.ejb.NoSuchEJBException: Bean has been deleted.
```

Solution

Restart Oracle Identity Manager server and retry upgrading the connector. This error may be encountered when Oracle Identity Manager is in idle state for a long time.

12

Managing Reconciliation

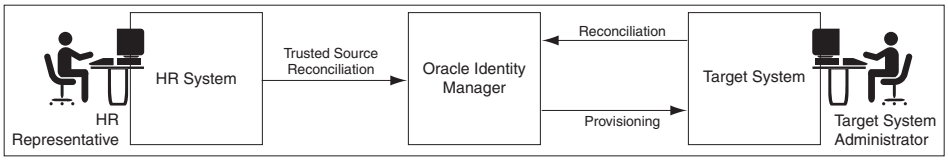
Reconciliation is the process by which operations, such as user creation, modification, or deletion, started on the target system are communicated to Oracle Identity Manager. The reconciliation process compares the entries in Oracle Identity Manager repository and the target system repository, determines the difference between the two repositories, and applies the latest changes to Oracle Identity Manager repository.

Reconciliation of roles, role memberships, and role hierarchy changes are handled as separate reconciliation events. Ideally role events must be submitted first and then only the membership events in order to avoid race conditions. For race conditions, the automatic retry logic allows the reconciliation engine to handle it.

> **See Also:** "Handling of Race Conditions" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about race conditions

Figure 12–1 shows that provisioning and reconciliation involve synchronization from Oracle Identity Manager to the target system or from the target system to Oracle Identity Manager. Provisioning and reconciliation enable the provisioning system to build the managed identities in the target system as well as replicate the managed identities as they already exist in the target system.

Figure 12–1 Provisioning and Reconciliation



In Figure 12–1, a user is created by the HR representative when a new employee joins. The user is reconciled to Oracle Identity Manager by trusted source reconciliation. When the user is created in Oracle Identity Manager, the account for the user is provisioned in the target system. In the target system, the target system administrator can make changes in the account, which must be reconciled to Oracle Identity Manager.

In terms of data flow, provisioning provides the outward flow from the provisioning system by using a push model, in which the provisioning system indicates the changes to be made to the target system. Reconciliation provides the inward flow into the provisioning system by using either a push or a pull model, by which the provisioning system finds out about any activity on the target system.

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 consequently because of changes occurring in the target system are managed by using the Event Management section in Oracle Identity System Administration, which addresses these event management needs. See "Managing Reconciliation Events" on page 12-10 for information about managing reconciliation events by using Oracle Identity System Administration.

This section consists of the following topics:

- Types of Reconciliation
- Managing Reconciliation Events

> **See Also:** "Customizing Reconciliation" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about reconciliation features and architecture

12.1 Types of Reconciliation

Reconciliation can be of different types, as shown in Table 12–1:

Table 12–1 Types of Reconciliation

Classification Criteria	Reconciliation Type
Object being reconciled	Based on identity being reconciled, such as user, account, role, organization, or relationship that includes role hierarchy and role membership
Mode of reconciliation	Changelog
	Regular
Approach used for reconciliation	Incremental reconciliation
	Full reconciliation

This section describes the following topics:

- Reconciliation Based on the Object Being Reconciled
- Mode of Reconciliation
- Approach Used for Reconciliation

12.1.1 Reconciliation Based on the Object Being Reconciled

Reconciliation depends on the entity object that is being reconciled. The following entities in Oracle Identity Manager are reconciled:

- **User:** A user is an identity that exists within and is managed through Oracle Identity Manager.
- **Account:** An account entity is granted to a user in Oracle Identity Manager. It represents a collection of the attributes and privileges for the user that uniquely identifies the user in a provisioning target. The existence of an account in Oracle Identity Manager makes it possible for the user to access the provisioning target.
- **Organization:** An organization entity represents a logical container of entities, such as users and other organizations, that exists in Oracle Identity Manager.

- **Role:** A role is a logical grouping of users to whom you can assign access rights within Oracle Identity Manager, provision resources automatically, or use in common tasks such as approval and certification.
- **Role hierarchy:** Role hierarchy is the inheritance of the parent role to child roles. The parent role has the same permissions and privileges on the members as the inherited roles.
- **Role membership:** Role membership means that the members of the inheritor role inherit from the inherited role. See "Managing Roles" in *Oracle Fusion Middleware Administering Oracle Identity Manager* for detailed information about membership and permission inheritance.

This section discusses the following topics:

- Trusted Source Reconciliation
- Account Reconciliation
- Reconciliation Process Flow

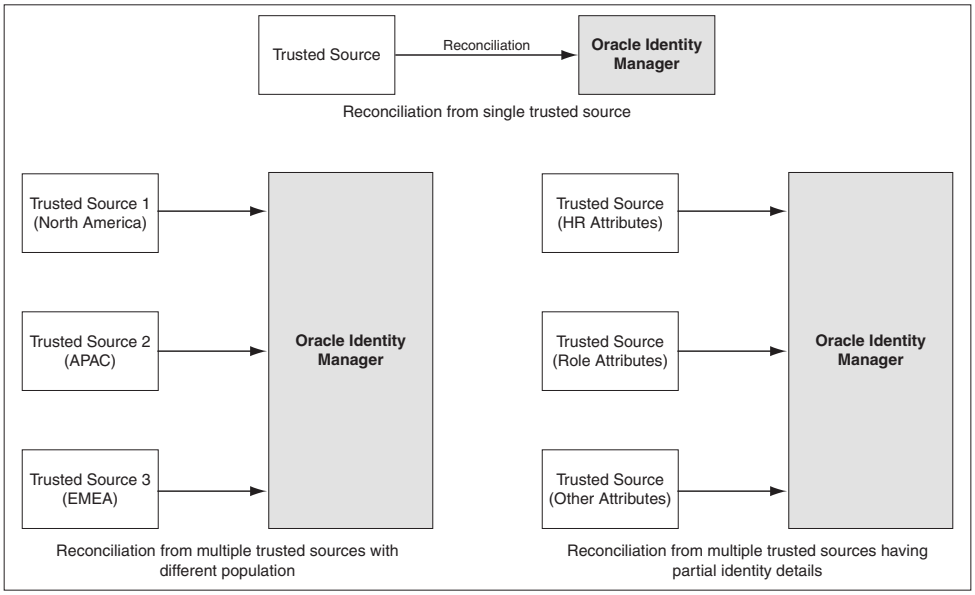
12.1.1.1 Trusted Source Reconciliation

If data is reconciled from a system that drives the *creation* of users, roles, role memberships, or role hierarchies in Oracle Identity Manager repository, then that reconciliation mode is called identity reconciliation, or authoritative source reconciliation, or trusted source reconciliation. The system that is being reconciled from is referred to as the authoritative source for the enterprise identities, and may be an HR system or a corporate directory.

Note: If the user login is not passed for trusted reconciliation, then the login handler generates the user login. The password is generated in postprocessing event handler, and notification is sent for the same.

As shown in Figure 12–2, the authoritative sources of identity may be more than one. The different authoritative sources may be the source of reconciliation for different categories of user identities or may be the source of reconciliation for different sets of attributes. The various events generated by the reconciliation engine are add, modify, and delete.

Figure 12–2 Trusted Source Reconciliation from Single and Multiple Authoritative Sources



In Figure 12–2, trusted source reconciliation from a single authoritative source and multiple authoritative sources are shown. Creation of user entities can be reconciled from multiple authoritative sources. In addition, different attributes can be reconciled from different multiple authoritative sources. For example, the user ID and e-mail ID can be provided by an authoritative source and role attributes can be provided by another authoritative source.

Trusted source reconciliation must be followed by account reconciliation when the target system is the source for identities as well as accounts. For instance, if Active Directory is the corporate LDAP repository in which user information is stored, then the user information is reconciled from the Active Directory target system. Subsequently, the Active Directory accounts are reconciled into Oracle Identity Manager by using a different connector. Identity reconciliation occurs only from trusted sources, by using connectors specific to those trusted sources.

Note: A reconciliation connector is a component developed to reconcile identities or accounts from a specific target system. Typically, a reconciliation connector is configured to be run as a scheduled task. However, there are push-based connectors, such as the PeopleSoft HR connector, for which there is no scheduled task to trigger the reconciliation.

12.1.1.2 Account Reconciliation

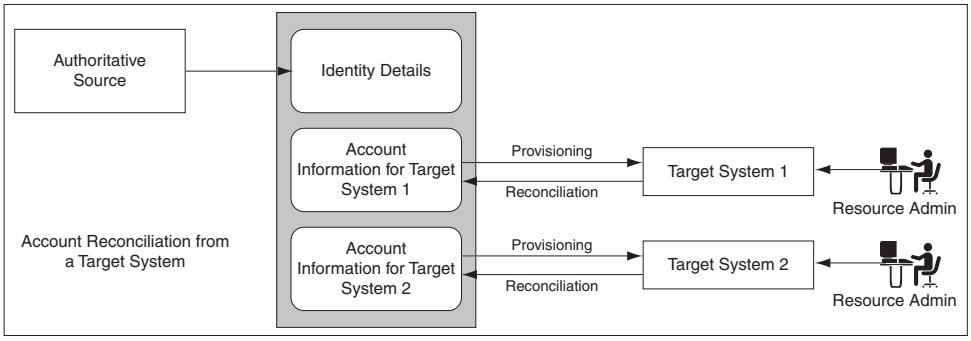
If the target system identities are accounts that get reconciled to Oracle Identity Manager, then that is target resource reconciliation or account reconciliation. This type of reconciliation is to reconcile a specific resource object that represents the target system being managed. There is always a corresponding provisioning flow for it. The identity retrieved from the target system maps to a resource object instance that has been provisioned to a user or organization.

Account reconciliation takes place in the following scenarios:

Scenario I

Identity gets created in Oracle Identity Manager from an authoritative source. The identities are provisioned with resources on the target system. Any change on the target system is reconciled with Oracle Identity Manager. Figure 12–3 shows account reconciliation from a target system:

Figure 12–3 Account Reconciliation From a Target System



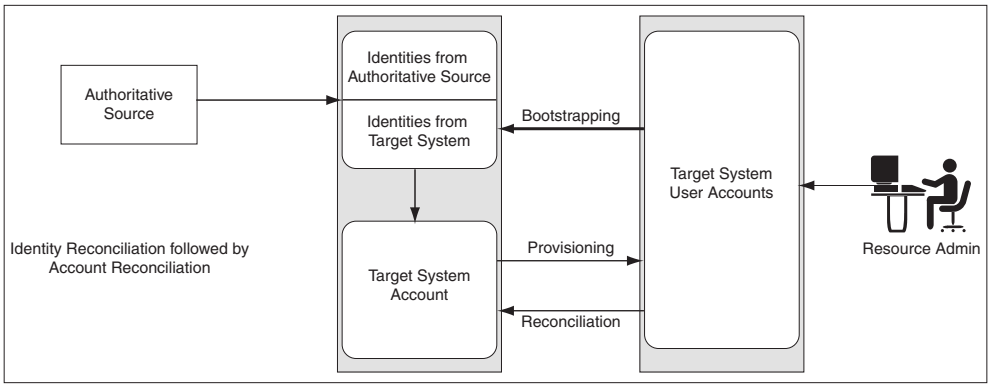
Scenario II

In this scenario, the target system initially plays the role of an authoritative source. Later it plays the role of a regular provisioning target. Following are the sequence of steps:

1. Identities are created in Oracle Identity Manager based on the target system entity details. Corresponding accounts are also created for these entities.
2. The entities are updated as provisioned entities in the target system.
3. The resource administrator at the target system makes changes to the accounts.
4. The changes made on the target system are reconciled with Oracle Identity Manager.

Figure 12–4 shows identity reconciliation followed by account reconciliation:

Figure 12–4 Identity and Account Reconciliation

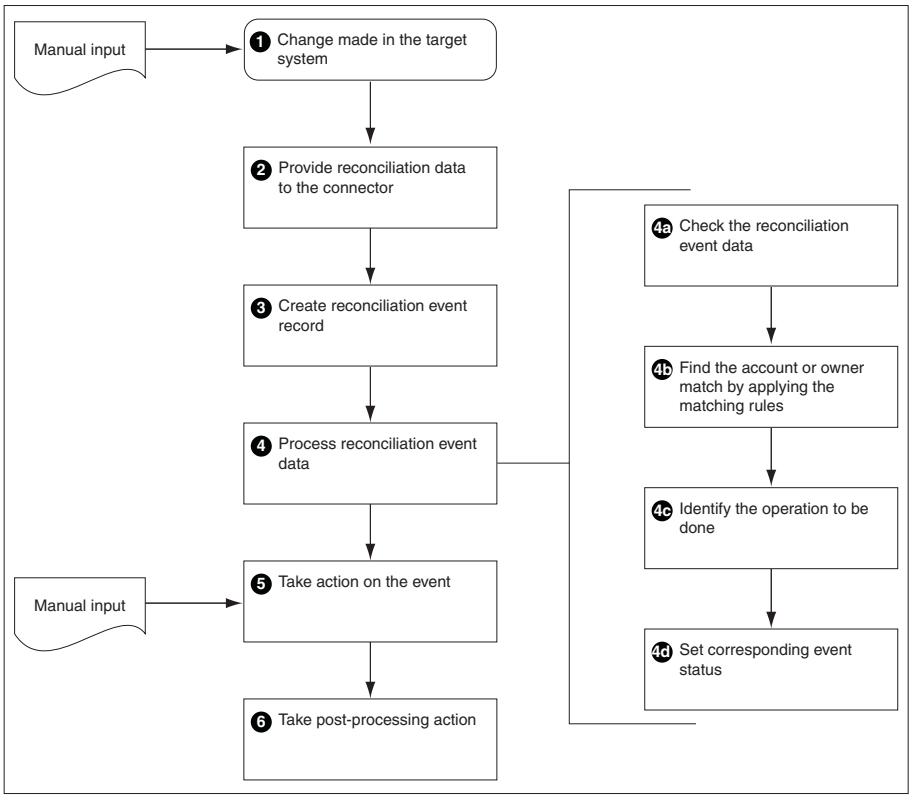


Note: When the value of the XL.UserProfileAuditDataCollection property is set to an audit data collection level, then the account reconciliation performs the matching in the database layer at a batch-level and performs the event action by using the provisioning APIs. This in turn triggers the audit event handlers for account reconciliation. By default, the value of this property is set to Resource Form. See "Administering System Properties" for information about system properties in Oracle Identity Manager in the *Oracle Fusion Middleware Administering Oracle Identity Manager*.

12.1.1.3 Reconciliation Process Flow

The reconciliation process flow is shown in Figure 12–5:

Figure 12–5 Reconciliation Process Flow



Reconciliation process involves the following steps:

1. **Changes in the target system:** The various activities that can happen in the target system are creation, modification, or deletion of user, account, role, role membership, or role hierarchy.

Note: If you create an entity on an external system and then modify it a short time later, reconciliation processes the create entity step, but the modify entity step fails with the Creation Failed event status. This is because reconciliation cannot process a create and a modify action for the same entity in the same batch process.

However, the entity modification action can be resubmitted for reconciliation at a later time by one of the following built-in mechanisms:

- The "Automated Retry of Failed Async Task" scheduled task will run to re-process the failed events without any manual intervention.
- The failed event is re-processed if the "Manual Retry Error Handling Mechanism" is triggered.

Reconciliation failure messages that are caused by processing conflicts within the same batch process should be regarded as transitory failures only.

2. **Providing reconciliation data:** When the creation, modification, or deletion event occurs, data about that event is sent to the reconciliation service by using reconciliation APIs.

   Note: Reconciliation service refers to the collection of reconciliation engine, reconciliation APIs, and the associated metadata and schema.

3. **Creation of reconciliation event record:** When the data for a reconciliation event is provided to reconciliation service, a record of that event is stored in Oracle Identity Manager repository.
4. **Processing of the reconciliation event data:** The data received is then evaluated to determine the actual operation to be performed in Oracle Identity Manager based on the changes in the target system. The evaluation involves application of a specific set of rules that help in:
   a. Identifying whether the data is for an account or for an identity that Oracle Identity Manager already has a record of
   b. Identifying the owner of the account or identity that the data represents
   c. Defining the context-sensitive action to be taken
   d. Setting the status of the event at the end of evaluation and the action that the reconciliation engine must take
5. **Taking action on the event:** Based on the evaluation result of processing the reconciliation event data, the intended action is taken. The various actions can be:

   Note: The actions on the event can be manually performed through the UI, or they can be automatic actions.

   - Creating a new account and associating with proper owner identity
   - Updating the matched account

- Deleting the matched account
- Creating a new user in Oracle Identity Manager
- Modifying an existing user in Oracle Identity Manager
- Deleting an existing user
- Enabling and disabling account status by updating the status attribute
- Enabling or disabling user
- Creating, updating, or deleting role
- Creating or deleting role membership
- Creating or deleting role hierarchy

See Also: "Reconciliation Engine" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about role membership and role hierarchy

6. **Follow up actions triggered by the reconciliation event:** After the action is taken, follow up tasks can be started based on the reconciliation event. An example of follow up tasks or post-processing task is creating a request to provision a resource, such as a laptop computer, after a user creation event.

12.1.2 Mode of Reconciliation

The mode of reconciliation is either pull or push that depends on the connector used. Most connectors, such as Active Directory, use the pull model. For the pull model, a pull reconciliation task is scheduled in the IAM Scheduler. The task runs at recurring intervals.

See Also: "Managing the Scheduler" on page 18-1 for information about the IAM Scheduler

Typically, the pull-based reconciliation connectors submit the reconciliation events within a scheduled task. Every time the scheduled task runs, a new reconciliation run is triggered and the reconciliation events are created in batches. When the batch size is met, the batch is submitted for processing. At the end of the scheduled task, an end of job listener is triggered, which submits all the batches whose size is not met.

Other reconciliation connectors, such as the PeopleSoft connector, use a push model. The connector comprises of an HTTP listener that detects any asynchronous messages issued by PeopleSoft. On receiving a message, the listener submits reconciliation events by calling the reconciliation API. The events are processed by the reconciliation engine in batches when the batch size is met. For batches where batch size is not met, a scheduled task runs periodically and submits the batches for reconciliation processing.

Pull or push model is used based on the nature of the target system and how the changes can be detected in the target system. But irrespective of the push or pull model being used, reconciliation is performed by using a scheduled task that runs in the IAM Scheduler.

Note: You can also create the reconciliation events directly by using the reconciliation APIs.

Changelog reconciliation is the default reconciliation mode. In this mode, only changed attributes are reconciled. Unspecified fields are ignored. You typically use the Changelog reconciliation mode when a connector is aware of the list of changed attributes. Along with the changed attributes, Oracle Identity Manager needs a list of required fields for matching. The Changelog reconciliation mode was supported in previous Oracle Identity Manager releases, so all connectors work in this mode.

Regular reconciliation is a new reconciliation mode, introduced in this release, where the reconciliation engine completely replaces the existing snapshot of the entity. You typically use this reconciliation mode when the connector cannot determine which attributes have changed, and therefore, sends an entire snapshot of the entity. For new connectors, you can specify this mode when performing a full reconciliation. Using regular reconciliation mode results in better performance because the events are processed faster.

Note: The mode of reconciliation depends on the connector implementation. For information about connector implementation, see "Connector for Reconciliation" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

Table 12–2 lists the differences between regular and changelog reconciliation modes:

Table 12–2 Regular and Changelog Reconciliation Modes

Regular	Changelog
Must pass a full set of mapped attributes	Must pass a subset of mapped attributes that are required by the specific profile and used by matching a rule
Performs better in batch processing mode (no difference in performance while in single event processing mode)	
Creates and updates all fields	Creates and updates only specified fields, and all other fields remain unchanged

See Also: "Changing the Profile Mode" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about changing the reconciliation mode

12.1.3 Approach Used for Reconciliation

When you run reconciliation for the first time on a target system, all users and accounts on the target system are reconciled into Oracle Identity Manager by default. This is called full reconciliation. To perform full reconciliation, the connector sends the reconciliation events for each entity in the target system. The reconciliation engine processes the events as create or update events depending on whether or not the entity already exists in Oracle Identity Manager. The connector also identifies all the deleted entries and sends the deletion events to Oracle Identity Manager.

At the end of full reconciliation, the connector typically sets the last execution time parameter to the time when the reconciliation run ends. For the next reconciliation run, only the entity records that have been added, modified, or deleted after the first reconciliation run ended are fetched for reconciliation. This is called incremental reconciliation.

You can manually switch from incremental reconciliation to full reconciliation by setting the value of the timestamp IT resource parameter to 0.

12.2 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.

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. These reconciliation runs are scheduled to run by using the Oracle Identity Manager Scheduler.

> **See Also:** ""Managing the Scheduler" on page 18-1" for detailed information about the scheduler

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

12.2.1 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

12.2.1.1 Performing a Simple Search for Events

To perform a simple search for events:

1. Login to Oracle Identity System Administration.
2. In the left pane, under Provisioning Configuration, click **Reconciliation**. The Advanced Administration is displayed with the Reconciliation section in the Event Management tab active.
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.

12.2.1.2 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 12–3 lists the fields in the Search: events page.

Table 12–3 Advanced Search Fields

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

Table 12–3 (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), delete, or changelog.
Profile Name	The name of the reconciliation profile this event pertains to. **See Also:** "Reconciliation Profile" in the *Oracle Fusion Middleware Developing and Customizing Applications 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, Current 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

12.2.2 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 12–4:

Table 12–4 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 12–5:

Table 12–5 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 12–5 (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.

12.2.3 Determining Event Actions

The list of actions allowed for an event depends on the status, type, and operation of the event. Table 12–6 lists the possible actions for each type and status of events.

Table 12–6 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.

12.2.4 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 **Reevaluate Event**. The Reevaluate Event dialog box is displayed with the event IDs that you have selected.
3. Click **Reevaluate**. 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 Reevaluate, then only the valid events are re-evaluated.
- All event actions are tracked in the Event History table.

12.2.5 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 **Close**. 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.

12.2.6 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

12.2.6.1 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. Click the lookup icon, and perform a user search.
3. Select a user from the search result, and click **Link**. A confirmation message is displayed that states that the ad hoc linking with the event is successful.

12.2.6.2 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.

12.2.6.3 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

12.2.6.3.1 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 12-16.

For information about manual linking, see "Manual Linking" on page 12-16.

12.2.6.3.2 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 12-16.

Part VI

Requests

This part describes the administration of request catalog and request service.

Part I contains the following chapter:

- Chapter 13, "Managing the Access Request Catalog"

13

Managing the Access Request Catalog

This chapter provides an introduction to the Access Request Catalog and describes the key features, benefits and use cases of the Access Request Catalog. It contains the following sections:

The Access Request Catalog provides a simple, intuitive, web-based user interface that allows business users to request access to roles, application instance, and additional access (also known as entitlements) within applications.

The Access Request Catalog allows a business to categorize and publish roles, application instance, and entitlements to the Catalog and provide additional business context using extensible metadata. Users use familiar request access for themselves using an intuitive "Catalog search" and "Shopping Cart" user experience.

13.1 Access Request Catalog

This section provides an introduction to the Access Request Catalog. It contains the following sections:

13.1.1 Access Request Challenges

Enterprises have tried to simplify and streamline the process of managing the identity lifecycle and access privileges of end users as part of improving operational efficiency and reducing IT costs. To meet these goals, businesses have tried to implement various solutions to allow end users to manage their own identity and access. However, they have faced several challenges in doing so:

- End-users had to be trained to understand IT concepts and terminology and use IT processes to request access.

- The training cycle had to be repeated as new employees joined, lowering productivity, and increasing IT costs.
- End-users had to get IT assistance when their requests were not fulfilled in a timely manner and did not have visibility into the status of their request.
- Typically, additional access within an application had to be granted by IT or by Application administrators.
- This limited business users' view of available access and limited their productivity, while forcing them to rely on IT.

The Access Request Catalog addresses these challenges by providing an easy to use web interface where users can search and browse various types of access and select the ones they need to perform their job duties. It provides the following benefits:

- The end user does not need to know technical jargon or follow IT processes to request access. The Catalog uses well-known and familiar search and shopping cart patterns to guide the user through the access request process.
- The end-user does not need to know specific application instance, role or entitlement names. The Catalog provides an extensible metadata model and provides tagging capabilities. This allow business users to specify alternate terms to be used to search for the specific access. End users can search the Catalog using combinations of keywords and wildcards to search for the access they need.

13.1.2 Concepts

The following discussion introduces key access request catalog concepts

- Catalog

  Catalog (aka Request Catalog) offers a consistent and intuitive request experience for customers to request Roles, Entitlements and Application Instances following the commonly used Shopping Cart paradigm. The catalog is a structured commodity with its own set of metadata.

- Catalog Item

  A Catalog Item is an item (Roles, Entitlements or Application Instances) that can be requested by a user, either for themselves or on behalf of other users.

- Category

  A Catalog Item Category is a way to organize the request catalog. Each catalog item is associated with one and only one category. A catalog item navigation category is an attribute of the catalog item. Catalog System Administrators can edit a Catalog Item and provide a value for the category.

  Note: You cannot leave Category field blank for a catalog item. Therefore, you must ensure that a value is present for the category.

- Application Instance

  An Application Instance represents an account on particular target. When users request an application instance, they are requesting an account in a particular target. Application Instances can be connected, if fulfillment is automated via a Connector, or disconnected, if fulfillment is manual. Application Instances can have entitlements associated with them.

- Enterprise Roles

Enterprise Roles are defined by customers. Enterprise Roles have policies associated with them. Users can request enterprise roles via the Catalog. When a role is granted, application instances or entitlements are provisioned to the user.

- Entitlement

  Entitlements are privileges in an application that govern what a user of the application can do.

- Catalog User-defined field

  Catalog User-defined fields are additional attributes that are added by customers to the Catalog entity

- Catalog Item Metadata

  Catalog Item Metadata refers to the values for the Catalog Item attributes. Metadata can be managed on a per-item basis by the Catalog System Administrator or can be populated in bulk.

- Tags

  Tags are search keywords. When users search the Access Request Catalog, the search is performed against the tags. Tags are of three types

  – Auto-generated: The Catalog synchronization process auto-tags the Catalog Item using the Item Type, Item Name and Item Display Name

  – User-defined: User-defined Tags are additional keywords entered by the Catalog System Administrator

- Catalog System Administrator

  The Catalog System Administrator is a global security role. The Catalog can be managed by members of this role only.

- Shopping Cart

  The Shopping Cart refers to the collection of Catalog Items that are being requested. A user can have only one cart active at any given time and the cart can contain roles, application instances, entitlements, or any combination of the three.

- Catalog synchronization

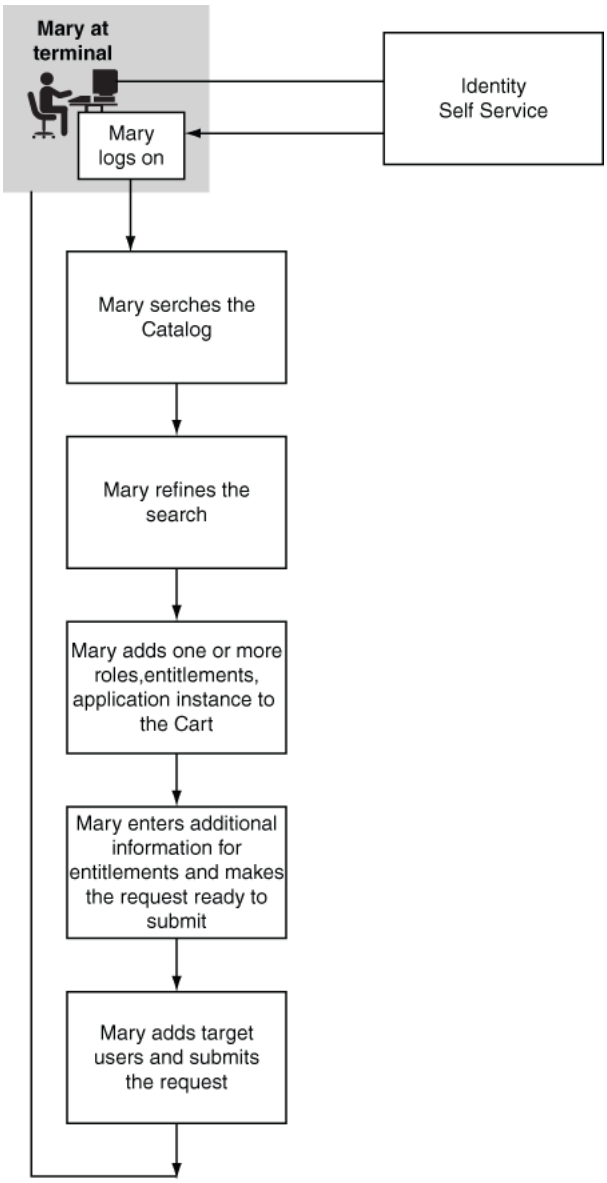
  Catalog synchronization refers to the process of loading roles, application instances, and entitlements into the Catalog.

13.1.3 Catalog Use cases

Use cases in this section explain how the access request catalog make it easy for end users to request roles, application instance, and entitlements required to perform their duties.

Requesting access

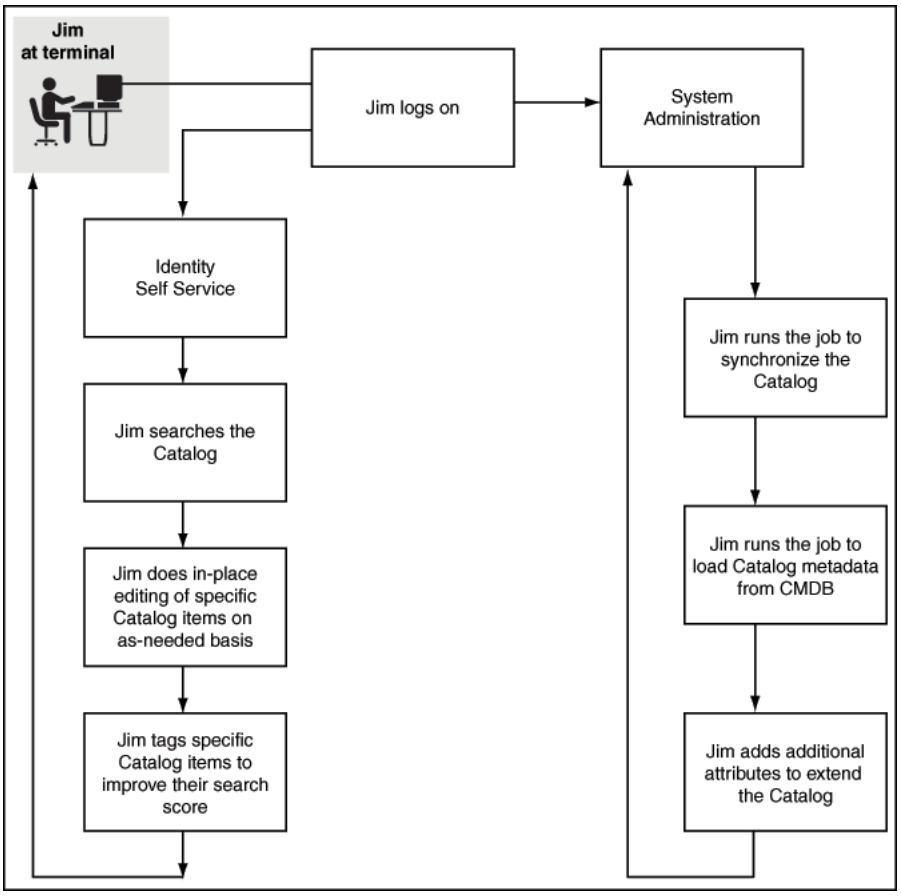
Mary, a Manager in MyCorp, would like to request access to MyCorp Trading application for herself and her directs. To do this, she searches the Catalog using the keyword trading. The catalog returns all items that match Mary's keywords and that she is allowed to request. Mary filters the search results by selecting Application from the list of categories. The Catalog returns a reduced set of search results. Mary adds the MyCorp Trading application to the cart and checks out. She adds herself and her directs to the request and submits the request.



Administering the Catalog

Jim, a Catalog System Administrator, would like to onboard new application instance and their entitlements, add additional attributes and improve the searchability of the catalog items. He runs the Catalog Synchronization Job scheduled job to harvest the new application instance and their entitlements. Next, he extends the Catalog metadata by adding additional attributes and identifies certain attributes as searchable. Next, he loads the catalog with metadata and tags for the new attributes. For certain Catalog items, he searches the Catalog and edits the Catalog item in place.

Note: The Catalog System Administrator must have the System Configuration Administrator admin role for running the Catalog Synchronization Job.



These use cases are typical examples of using the Access Request Catalog to make applications and entitlements in the applications and roles visible in the Catalog and allowing users to request access to them via simple web-based interface.

13.2 About the Access Request Catalog

This section covers the features and benefits of the Access Request Catalog and its architecture. It contains the following topics

- Section 13.2.1, "Features and Benefits"
- Section 13.2.2, "Architecture"

13.2.1 Features and Benefits

The Access Request Catalog is a searchable, categorized collection of entities that are requestable in Oracle Identity Manager. Any authenticated user can access the Catalog and search the Catalog using one or more keywords and search operators, add one or more Catalog items into a shopping cart and submit a request for themselves and others.

Key features of the access request catalog include:

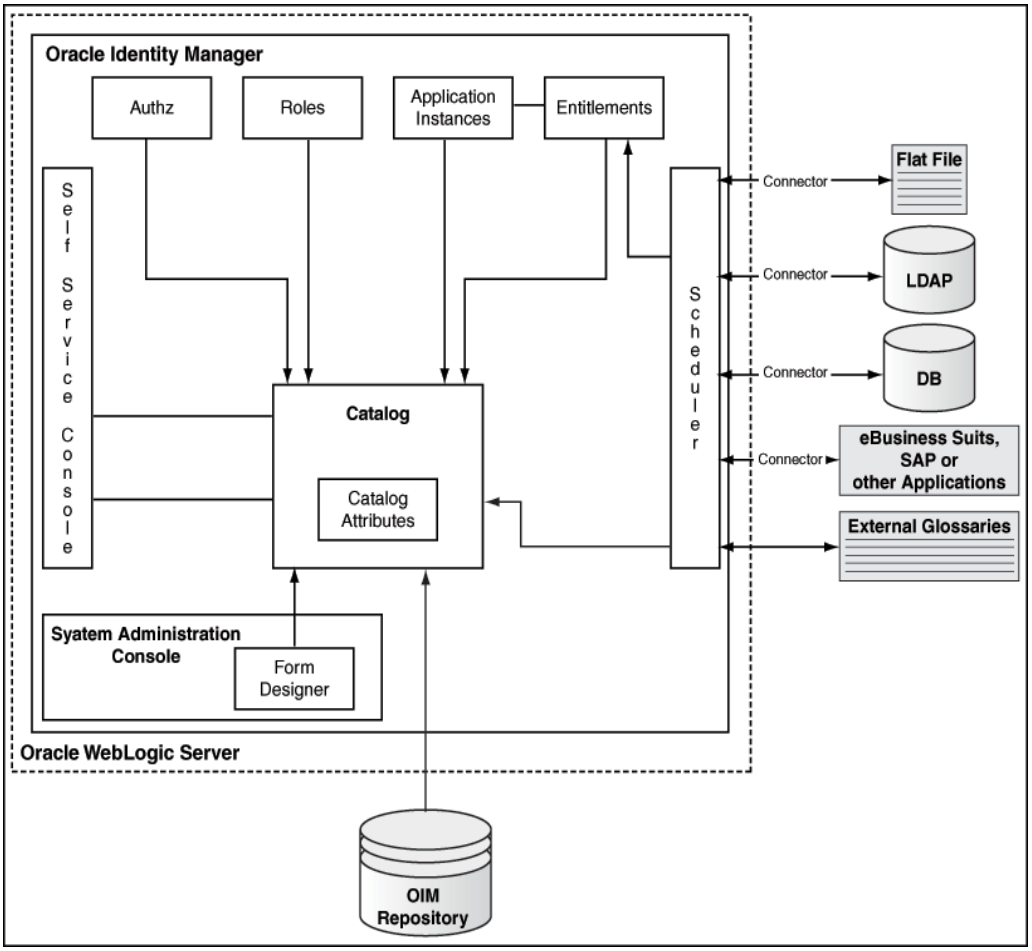
- Extensible Catalog schema that allows administrators to add additional attributes and specify how the attribute is rendered using a simple browser-based UI
- Automated harvesting of roles, applications, and entitlements

- Automated loading of Catalog metadata using a CSV file
- Powerful search using keywords with support for complex search operators
- Flexible categorization model that allows the Catalog to be organized based on customer choice
- Catalog search results secured based on viewer privileges of the requester
- Catalog item data available via a web service for use in workflows

13.2.2 Architecture

Figure 13–1 shows the components of the Access Request Catalog and its relationship with other components of Oracle Identity Manager.

Figure 13–1 High-Level Catalog Architecture



The Access Request Catalog consists of the following components:

1. Catalog Tables
2. Catalog Loaders
3. Catalog Metadata
4. Catalog User Interface in the Identity Self Service Console

13.3 Configuring the Access Request Catalog

This section describes the following configurations for the access catalog:

- Section 13.3.1, "Adding More Attributes to the Default Search Form"
- Section 13.3.2, "Configuring Application Selection Limit in Entitlement Search"
- Section 13.3.3, "Configuring Catalog to Use a Custom Search Form"

13.3.1 Adding More Attributes to the Default Search Form

Additional attributes can be added to the catalog search form. The attributes marked as searchable are displayed automatically as text fields in the default search form. These attributes must be added to the cart details form via customization. For information about defining a custom attribute, see "Configuring Custom Attributes" on page 7-1.

13.3.2 Configuring Application Selection Limit in Entitlement Search

You can configure the number of applications that can be selected in the default search form during entitlement search. This limit is configurable by using the `Catalog Advanced Search Maximum Applications` system property. For information about this system property, see "System Properties in Oracle Identity Manager" on page 20-1.

13.3.3 Configuring Catalog to Use a Custom Search Form

For advanced customizations to the catalog search, the default catalog search form can be replaced with a custom-built search form. The catalog search form can be configured by using the `Catalog Advanced Search Taskflow` system property. For information about developing a custom taskflow for catalog search, see "Customizing the Catalog Search Form" in *Developing and Customizing Applications for Oracle Identity Manager*.

13.4 Administering the Access Request Catalog

This section describes the basic administration of the Access Request Catalog. It consists of the following topics

- Section 13.4.1, "Pre-requisites"
- Section 13.4.2, "Common Tasks"
- Section 13.4.3, "Configuring Catalog Auditing"
- Section 13.4.4, "Configuring Hierarchical Attributes of Entitlements"
- Section 13.4.5, "Database Best Practices for Access Request Catalog"

13.4.1 Pre-requisites

The Access Request Catalog is used by end-users to request access to roles and entitlements to help them perform their duties. As a result, it is very important that the Catalog be current, have a rich metadata and be organized so that users can find the right access. To ensure this, you need to have a plan to manage the Access Request Catalog. The ensuring sections give the steps that you should follow to administer the Catalog. Before implementing those steps, there are certain pre-requisites. These include

- Section 13.4.1.1, "Setting up the Catalog System Administrator"

- Section 13.4.1.2, "Defining the Catalog Metadata"

13.4.1.1 Setting up the Catalog System Administrator

The Catalog System Administrator is an admin role, similar to the System Administrator and System Configurator role. In Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0), a member of this role (and those of the System Administrators role) can perform the following actions:

- Load the Catalog
- Manage Catalog Items
- Manage Request Profiles

This role is a global role and not scoped by organization.

To grant the Catalog System Administrator:

1. Log in to Oracle Identity Self Service.
2. Click the **Manage** tab, and click **Organizations**.
3. Search and open the Top organization.
4. Click the **Admin Roles** tab.
5. Select the **Catalog System Administrator** admin role and click **Assign** in the toolbar.
6. Search and select the users that you want to assign, and click **Add Selected.**
7. Click **Add** to add the users.

The new members of the Catalog System Administrator role can login to the Self Service Console and start managing the Catalog.

13.4.1.2 Defining the Catalog Metadata

A rich catalog metadata is important to for the following reasons:

- End-users are only interested in getting access to what they need to perform their job duties. When they search and browse the Catalog, the information presented to them must relate to the business. If the Catalog is sparse (minimal attributes), users will not know which access to pick. If the Catalog is rich but technical, users will get confused and will choose not to use the Catalog.
- Requesters and Approvers need as much contextual information as possible to help them submit a request or approver one. When approvers review a request, the Catalog item detail helps them understand what is being requested, why and the impact of approving the request.
- Approval workflows use routing rules to correctly determine approvers. These rules need access to additional context about the requested item to do approver resolution. If the Catalog information is sparse, the routing rules will not have enough data available to determine the correct approvers.

To meet these challenges, the Catalog must contain additional metadata that can help place the access, that is the Catalog item, in the correct business context.

To add one or more attributes to the Catalog:

1. Log in to the Oracle Identity System Administration Console.
2. Create and activate a sandbox. See "Managing Sandboxes" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

3. Under System Entities, click **Catalog**.
4. Click **Custom New Attribute** to add an attribute.
5. Select from one of the pre-defined attribute types and click **OK.**
6. Provide the necessary information and click **Save and Close.**

> **Note:** If new custom attribute (UDF) is made Searachable, it is recommended to create a normal index on the database column of the custom attribute for optimal search performance.You can find the database columns of custom attributes in CATALOG table of Oracle Identity Manager schema.

7. Add additional attributes as required.

   You have completed the first step in extending the Catalog.
8. If you do not want to modify the Catalog search results or Catalog Item details UI, then you can have your changes reviewed and after approval of the changes, export and publish the sandbox. It is recommended that you export the sandbox to store all the changes made in your sandbox.

   If you want to modify the Catalog search results and Catalog Item details UI, then proceed further.
9. Logout and login to the Identity Console as a member of the System Administrator role.
10. Create a new sandbox and activate it.
11. Add the attribute to the catalog details page by referring to Section 7.5, "Adding a Custom Attribute".
12. Export and publish the sandbox.

13.4.2 Common Tasks

This section describes the common tasks to be performed by the Catalog System Administrator. It consists of the following tasks:

- Section 13.4.2.1, "Onboard Applications and Roles"
- Section 13.4.2.2, "Bootstrapping the Catalog"
- Section 13.4.2.3, "Ongoing Synchronization"
- Section 13.4.2.4, "Enriching the Catalog"
- Section 13.4.2.5, "Managing Catalog Items"

13.4.2.1 Onboard Applications and Roles

The Access Request Catalog must be populated with enterprise roles, application instances and entitlements so that users can search and request for access. You must develop a process by which enterprise roles, application instances and entitlements can be on-boarded to the Catalog with minimal administrator intervention. This section covers the various steps involved in on-boarding roles, application instances and entitlements into the Catalog.

13.4.2.1.1 Prepare an Onboarding checklist Use the following oboarding checklist items to develop a high-level process for onboarding roles, application instances and

entitlements into the Access Request Catalog. Later, you can follow individual checklists for roles, application instances, and entitlements.

- Identify Catalog System Administrators
- Identify and extended Catalog attributes
- Customize Catalog search results UI
- Customize Catalog Item Details UI
- Identify navigational categories
- Identify Owners, Certifiers, Approvers for roles and applications
- Identify sources of truth for Catalog Item metadata/glossary
- Develop procedures to generate and load Catalog item metadata/glossary
- Develop glossary of tags and a process to maintain tags

13.4.2.1.2 Onboarding Roles There are no onboarding steps for enterprise roles. Roles, belonging to a role category other than Oracle Identity Manager Roles are published directly to the Catalog when they are created.

When user edits the role and changes its category from Oracle Identity Manager Role to any other category, then the Catalog Synchronization Job scheduled job must be run to have the role searchable in the catalog.

13.4.2.1.3 Onboarding Application Instances Application Instances require additional configuration before they can be requested by end users. Use the following checklist items to make sure that you have performed the configuration required to onboard application instances:

- Ensure that the Connector is installed (for new targets)
- If you are upgrading Oracle Identity Manager from Release 9.1.*x* or 11*g* Release 1 to 11*g* Release 2 (11.1.2.3.0), see "Upgrading Oracle Identity Manager 11g Release 1 (11.1.1.x.x) Environments" of the *Oracle Fusion Middleware Upgrade and Migration Guide for Oracle Identity and Access Management* for information about mandatory post-upgrade steps
- Verify that the process forms have an IT resource field
- Verify that you have defined the form field properties correctly
- Verify that you have created the application instances with suitable display names and descriptions
- Verify that you have created the forms required for account requests
- Verify that you have published the application instances to the relevant organizations
- For disconnected applications, verify that you have created the application instances. See "Managing Disconnected Resources" on page 10-25 for detailed description of the steps

After verifying the steps in the check list, follow the instructions below to onboard application instances.

> **See Also:** Section 10.2, "Managing Application Instances" for more information on managing Application Instances

Steps to onboard Application Instances

1. Login to the Identity System Administration as a member of the System Administrator role.
2. In the left pane, under System Configuration, click **Scheduler**.
3. Search for the Catalog Synchronization Job scheduled job.
4. Check the `Process Application Instances` parameter.
5. Set the parameter `Mode` to Incremental.

13.4.2.1.4 Onboarding Entitlements Use the following checklist items to make sure that you have performed the configuration required to onboard entitlements.

> **Note:** Job entitlement list loader should be executed before executing the Catalog Synchronization Job scheduled job.

- Ensure that the Connector is installed (for new targets)
- If you are upgrading Oracle Identity Manager from Release 9.1.x or 11g Release 1 to 11*g* Release 2 (11.1.2.3.0), see "Upgrading Oracle Identity Manager 11g Release 1 (11.1.1.x.x) Environments" of the *Oracle Fusion Middleware Upgrade and Migration Guide for Oracle Identity and Access Management* for information about mandatory post-upgrade steps
- Verify that the process forms have an IT resource field
- Verify that you have defined the form field properties correctly
- Verify that you have correctly associated the parent and child forms
- Verify that you have run the common lookup reconciliation job for ICF-based targets
- Verify that you have run the connector-specific lookup reconciliation jobs for non-ICF connectors
- Verify that you have created application instances correctly, corresponding to the resource object and IT resource instance specified in the Lookup Reconciliation job
- Verify that you have published entitlements to relevant organizations
- Verify that you have run the entitlement list loader job, so that data can be populated in ent_list table

After verifying the steps in the check list, follow the instructions below to onboard entitlements

Steps to onboard Entitlements

1. Login to Identity System Administration as a member of the System Administrator role.
2. In the left pane, under System Configuration, click **Scheduler**.
3. Search for the Catalog Synchronization Job scheduled job.
4. Check the `Process Entitlements` parameter.
5. Set the parameter `Mode` to `Incremental`.

Note:

- If its a first time harvesting, then you should set the parameter to `Full`.
- If the parameter mode is Incremental, then only those entities are picked by scheduled task for processing, whose create date is greater than update date for creation, and update date is greater than update date value.

13.4.2.2 Bootstrapping the Catalog

Bootstrapping refers to the process of populating the Catalog for the first time. After Bootstrapping large number of any entity, you can gather statistics on base tables. This section refers to bootstrapping the Catalog after you have installed Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0). If you are upgrading from Oracle Identity Manager 9.1.x or 11g Release 1, then see Chapter, "Upgrading Oracle Identity Manager 11g Release (11.1.1.5.0) Environments" of the *Oracle Fusion Middleware Upgrade and Migration Guide for Oracle Identity* and Access Management.

Pre-requisites

- You have extended the Catalog using the Catalog system entities by following the steps given in Section 13.4.1.2, "Defining the Catalog Metadata".
- You have carried out the necessary UI customization steps required when a user-defined field is added to the Catalog.

13.4.2.2.1 Bootstrapping the Catalog with Roles There are two ways to bootstrap the Catalog with Roles.

- Bootstrapping the Catalog with Roles when you are not using Oracle Identity Analytics customer

  In Oracle Identity Manager 11g R2, roles are published immediately to the Catalog when they are created and assigned a role category other than the Oracle Identity Manager Roles category. If you have made changes to the role categories or need to synchronize the enterprise roles with the Catalog, follow the steps given below

  To bootstrap the catalog with roles:

  1. Login to Identity System Administration as a member of the System Administrator role.
  2. In the left pane, under System Configuration, click Scheduler.
  3. Search for the Catalog Synchronization Job scheduled job.
  4. Check the `Process Roles` parameter.
  5. Set the parameter `Mode` to `Full`.

     Note: If you are running the job for the first time and the **Mode** is set to **Full,** then you must not provide any value in the Update Date parameter.

  6. Click **Run Now** to run the job immediately or provide a date and time to run the job later.

- Bootstrapping the Catalog with Roles when you are using Oracle Identity Analytics for managing the lifecycle of enterprise roles

13.4.2.2.2 Bootstrapping the Catalog with Application Instances Bootstrapping the Catalog with Application Instances requires additional steps to be carried out. Use the checklist given in Section 13.4.2.1.3, "Onboarding Application Instances" to ensure that you have completed the pre-requisites.

Once you have completed the pre-requisites, follow the steps given below to onboard application instances:

1. Login to Identity System Administration as a member of the System Administrator role.
2. In the left pane, under System Configuration, click **Scheduler**.
3. Search for the Catalog Synchronization Job scheduled job.
4. Check the `Process Application Instances` parameter.
5. Set the parameter `Mode` to `Full`.

> **Note:** If you are running the job for the first time and the **Mode** is set to **Full,** then you must not provide any value in the Update Date parameter.

6. Click **Run Now** to run the job immediately or provide a date and time to run the job later.

13.4.2.2.3 Bootstrapping the Catalog with Entitlements Bootstrapping the Catalog with Entitlements requires additional steps to be carried out. Use the checklist given in Section 13.4.2.1.4, "Onboarding Entitlements" to ensure that you have completed the pre-requisites.

Once you have completed the pre-requisites, follow the steps given below to onboard entitlements.

1. Login to Identity System Administration as a member of the System Administrator role.
2. In the left pane, under System Configuration, click **Scheduler**.
3. Search for the Catalog Synchronization Job scheduled job.
4. Check the `Process Entitlements` parameter.
5. Set the parameter `Mode` to `Full`.

> **Note:** If you are running the job for the first time and the **Mode** is set to **Full,** then you must not provide any value in the Update Date parameter.

6. Click **Run Now** to run the job immediately or provide a date and time to run the job later.

13.4.2.3 Ongoing Synchronization

To automate the process of onboarding roles, application instances, and entitlements, you can configure the Catalog Synchronization Job scheduled job in the following manner.

1. Login to Identity System Administration as a member of the System Administrator role.
2. In the left pane, under System Configuration, click **Scheduler**.
3. Search for the Catalog Synchronization Job scheduled job.
4. Check the `Process Roles`, `Process Application Instances`, and `Process Entitlements` parameters.
5. Set the parameter `Mode` to Incremental.
6. Provide a date and time to run the job later.
7. Set the Job frequency to run every five minutes.

13.4.2.4 Enriching the Catalog

Enriching the Catalog refers to the process of populating the Access Request Catalog with data so that the information is available for end-users to see. The additional data helps end-users understand the business context associated with the Catalog Item. The additional data is also available as part of the approval workflow, allowing the workflow to make intelligent routing decisions based on the data about the Catalog Item.

There are two ways to enrich the Catalog:

- Section 13.4.2.4.1, "Editing a Catalog Item Online"
- Section 13.4.2.4.2, "Enriching the Catalog in bulk from external sources"

Pre-requisites

- You have extended the Catalog using the Catalog system entities by following the steps given in Section 13.4.1.2, "Defining the Catalog Metadata".
- You have added UI customizations required when a user-defined field is added to the Catalog. See Chapter 7, "Configuring Custom Attributes" for information about adding user-defined fields and customizing the UI to display the user-defined field in the UI.
- You have created a Catalog System Administrator role and assigned users as given in Section 13.4.1.1, "Setting up the Catalog System Administrator"

13.4.2.4.1 Editing a Catalog Item Online To edit a Catalog Item online, using the Oracle Identity Manager Self Service Console:

> **Note:** Name, Display Name, and Description cannot be edited on the catalog screen. These are base level attributes and you cannot edit from Catalog UI.
>
> When editing a Catalog Item, for list of values (LOV) type of fields, it is recommended to select and specify values by picking from the associated lists, instead of typing the values into the fields directly.

1. Log in to Identity Manager Self Service as a member of the Catalog System Administrator role.
2. Click **Catalog** to access the request catalog.
3. Enter one or more keywords and click **Search**.
4. Use the Refine Search to find the Catalog Item(s) to be edited.
5. Select the Catalog Item to be edited.
6. In the Detailed Information section, edit the Catalog Item and click **Apply**. Verify the confirmation message.

13.4.2.4.2 Enriching the Catalog in bulk from external sources

While Catalog System Administrators can make use of the robust Catalog Item editing capabilities in the Oracle Identity Manager Self Service Console, there are scenarios where the data needs to be loaded in bulk from external sources.

Examples of bulk updates:

- MyCorp wants to provide users with asset information from their IT CMDB system or from their Corporate Asset Management system. The information cannot be entered manually since the CMDB or AMS system gets updated on a regular basis. In such a scenario, MyCorp needs a way to update the Catalog in bulk.
- MyCorp was using a home grown access request application prior to implementing Oracle Identity Manager 11g R2. This application contains the glossary and other relevant information about the roles, application instances and entitlements. As part of migrating to Oracle Identity Manager 11g R2, MyCorp Catalog System Administrators would like to move the Catalog Item information from the legacy system.

13.4.2.4.3 Loading data from an external source

Follow the steps given below to load data from an external source into the Catalog:

1. Export the data to be loaded into a comma-separated values format file.
2. Ensure that the first line of the file contains the Catalog attribute names.
3. Move the file to a file system that is accessible from the server on which is Oracle Identity Manager is deployed.
4. Login to Identity System Administration as a member of the System Administrator or System Configurator role.
5. In the left pane, under System Configuration, click **Scheduler**.
6. Search for the Catalog Synchronization Job scheduled job.
7. Provide the full path to the file in the parameter `File Path`.
8. Set the value of the parameter `Mode` to `Metadata`. Table 13–1 provides sample parameter details.

Table 13–1 Catalog Metadata Loader Sample

Parameter	Value
ENTITY_TYPE	Role
ENTITY_KEY	12
ENTITY_NAME	test

Table 13–1 (Cont.) Catalog Metadata Loader Sample

Parameter	Value
IS_REQUESTABLE	1
USER_DEFINED_TAGS	UDTags
CATEGORY	mycategory
AUDIT_OBJECTIVE	AO111
APPROVER_USER	1
APPROVER_ROLE	1
FULFILLMENT_USER	1
FULFILLMENT_ROLE	1
CERTIFIER_USER	1
CERTIFIER_ROLE	1
ITEM_RISK	5
CERTIFIABLE	1
STUDF	1

9. Click **Run Now** to run the job immediately, or select a date and click **Apply** to run the job later.

13.4.2.5 Managing Catalog Items

This section contains the following topics

- Deleting a Catalog Items of Type Roles
- Deleting Catalog Items of Type Application Instances
- Deleting Catalog Items of type Entitlements

13.4.2.5.1 Deleting a Catalog Items of Type Roles To delete role Catalog Items:

1. Login to Identity Self Service.
2. Search for the role to be deleted and delete the role.
3. The associated Catalog Item are marked as soft-deleted and will not appear in the Catalog.
4. For deleting large number of roles, use the APIs to delete the role. It is not recommended to use database techniques to delete roles.

13.4.2.5.2 Deleting Catalog Items of Type Application Instances Application Instances, in almost all use cases, represent a target system (sometimes known as an endpoint) and an account in a target system. When you delete an Application Instance, you are essentially decommissioning the target system from Oracle Identity Manager. Depending upon the scale of your deployment and the number of accounts provisioned to the target system, deleting an Application Instance can have a significant impact to the end users and their access.

To delete application instance Catalog Items:

1. Login to Oracle Identity System Administration.
2. Click **Application Instances**.

3. Search for application instances.
4. Select one or more application instances. Delete and confirm.
5. Click **Scheduler**.
6. Search for the Catalog Synchronization Job scheduled job.
7. Set the Mode to Incremental.
8. Click **Run Now** to run the job immediately or set it up to run at a particular time.

See "Deleting Application Instances" on page 10-9 for more information about deleting application instances.

13.4.2.5.3 Deleting Catalog Items of type Entitlements To delete entitlement Catalog Items:

1. To delete Entitlements, login to Oracle Identity System Administration.
2. Click **Lookups**.
3. In the Code column, enter the name of the Lookup Definition that contains the entitlement. Refer to the Connector documentation to find out the name of the Lookup Definition.
4. Delete one or more entitlement values.
5. Click **Scheduler**.
6. Search for the **Entitlement List Load j**ob.
7. Click **Run now**.
8. Search for the Catalog Synchronization Job scheduled job.
9. Set the Mode to **Incremental**.
10. Click **Run Now** to run the job immediately or set it up to run at a particular time.

13.4.3 Configuring Catalog Auditing

Catalog auditing maintains a footprint of changes in the access request catalog. By enabling catalog auditing, you can track who changes what and when in the access request catalog through the UI.

Catalog auditing stores the footprints of the following changes in the access request catalog:

- A change in the value of a catalog UDF.
- Any value of a catalog item attribute is changed from the catalog UI or any other custom UI.
- Following is the list of consolidated catalog attributes that are part of auditing during updation of catalog item:

  Category, Audit Objective, Approver User, Approver Role, Fulfillment User, Fulfillment role, Certifier User, Certifier Role, Item Risk, Certifiable

> **Note:** Auditing takes place only for those entities that can be modified through the Catalog UI. Audit does not happen for entities that are modified in the catalog through synchronization. In addition, auditing is not supported for User Defined Tags.

To configure catalog auditing:

1. Login to Oracle Identity System Administration.
2. Under System Configuration, click **Configuration Properties**.
3. Search for the Catalog Audit Data Collection system property with keyword XL.CatalogAuditDataCollection. The default value of this property is `none`, which specifies that catalog auditing is disabled.
4. Set the value of the XL.CatalogAuditDataCollection system property to `catalog`. This enables catalog auditing.
5. Click **Save**.

After enabling catalog auditing, the changes in the access request catalog are audited. For changes in the access request catalog, such as changing the risk level of a role, the footprints of the changes are stored in the CPA_CATALOG and CPA_CATALOG_FIELDS tables in the database on running the Issue Audit Massages Task scheduled job. For information about this scheduled job, see "Predefined Scheduled Tasks" in the *Oracle Fusion Middleware Administering Oracle Identity Manager*.

13.4.4 Configuring Hierarchical Attributes of Entitlements

You can enable the display of hierarchical attributes of entitlements to requesters, approvers, and certifiers to view additional details of entitlements (hierarchical attributes) in the catalog detail screen. The additional details of entitlements is called technical glossary. The technical glossary is displayed in a list view with bread crumbs at the top showing the navigational path. For information about viewing the additional details in the catalog detail screen, see "Viewing Hierarchical Attributes of Entitlements" in *Performing Self Service Tasks with Oracle Identity Manager*.

Note: The child entitlements are not requestable in the access catalog. The hierarchical entitlements feature is meant for display purpose only.

The additional details or hierarchical attributes is read-only information. This information must be provided in the form of an XML, which is seeded in Oracle Identity Manager. The technical glossary is inserted and replaced in the database. The following is a sample XML code of the hierarchical attributes:

```
<oim>
      <applicationInstances>
            <applicationInstance>SampleEBS</applicationInstance><!-- Application
Name for which entitlements are seeded-->
      </applicationInstances>
      <attributes>
            <attribute name="Responsibility Name"><!-- Label name of the field
which is marked Entitlement field in Child form-->
                  <entitlementValues>
                        <entitlementValue><!-- Below is the Hierarchical data XML
for Entitlement and Entitlement Display Name is used to denote entitlement -->
<value>Payables Menu</value>
<attributes>
      <attribute name="Menu">
            <entitlementValues>
                  <entitlementValue>
<value>ALR_OAM_NAV_GUI_USER_NAME</value>
<description>Alerts Manager View</description>
```

```
<attributes>
      <attribute name="Function Code">
            <entitlementValues>
                  <entitlementValue>
      <value>ALR_OBJ_ACTIVATE_ACCT</value>
      <description>Create, Activate, Deactive User Account</description>
</entitlementValue>
<entitlementValue>
<value>ALR_OBJ_EDIT_FORM</value>
</entitlementValue>
<entitlementValue>
<value>ALR_OBJ_VIEW_PERSON</value>
                  </entitlementValue>
            </entitlementValues>
      </attribute>
</attributes>
                  </entitlementValue>
                  <entitlementValue>
<value>EMPLOYEE_W2_MENU</value>
<description>Alerts Manager View</description>
<attributes>
      <attribute name="Function Code">
            <entitlementValues>
<entitlementValue>
      <value>Employee_OBJ_ACTIVATE_ACCT</value>
      <description>Create, Activate, Deactive User Account</description>
</entitlementValue>
<entitlementValue>
      <value>Employee_OBJ_EDIT_FORM</value>
</entitlementValue>
<entitlementValue>
      <value>Employee_OBJ_VIEW_PERSON</value>
</entitlementValue>
            </entitlementValues>
      </attribute>
                        </attributes>
                  </entitlementValue>
                  <entitlementValue>
<value>VISION_OAM_NAV_GUI</value>
<description>Alerts Manager View</description>
<attributes>
                        </attributes>
                  </entitlementValue>
            </entitlementValues>
      </attribute>
                              </attributes>
                        </entitlementValue>
                  </entitlementValues>
            </attribute>
      </attributes>

</oim>
```

RDBMS features, such as Securefile LOB and Oracle XML DB, are used for storing hierarchical data in Oracle Database. Securefile is a new re-architecture featuring entirely new disk formats, network protocol, space management, redo and undo formats, buffer caching, and intelligent I/O subsystems. It delivers substantially improved performance along with optimized storage for unstructured data, which resides in Oracle Database as compared to LOB's storage structure. Oracle XML DB provides a high-performance, native XML storage and retrieval technology. It absorbs

the W3C XML data model into the Oracle Database, provides new standard access methods for navigating and querying XML, and provides the advantages of relational database technology together with the advantages of XML.

To enable the display of additional details of the entitlements in the access request catalog:

1. Seed the additional hierarchical data in Oracle Identity Manager. To do so, create a XML file per the XSD with all the additional details about the entitlement. The XSD is used to register XML schema in the database.
2. Place the XML file in a directory in the Oracle Identity Manager server. You must have read and write permissions on the directory.
3. Specify the details of the technical glossary in the Catalog Synchronization Job scheduled job. To do so:
   a. Login to Oracle Identity System Administration.
   b. Under System Configuration, click **Scheduler**.
   c. Search and open the Catalog Synchronization Job scheduled job.
   d. In the Parameters section, in the Mode field, enter `Technical Glossary`.
   e. In the File Path field, enter the directory path of the XML file.
   f. Click **Apply**.

When you run the Catalog Synchronization Job scheduled job, a new link, which is called technical glossary details, is displayed just before the catalog details link for entitlements. Clicking this links opens the technical glossary additional information in a different tab. The XML file is deleted from the directory after processing and is moved to the archive directory with time stamp appended to its name.

Any failed record is logged in a file, which is placed in the xmlprocessedlogs directory. The log file has the name of the XML file with time stamp appended to it.

13.4.5 Database Best Practices for Access Request Catalog

Access Request Catalog uses "Oracle Text" option in Oracle database for text search capabilities. Oracle Text is a fast and accurate full-text retrieval technology integrated with Oracle Database.

The CATALOG table which contains catalog items is indexed using CONTEXT index type of Oracle Text. Although Oracle Text index operates like a regular database index, the architecture and processing behind Text index highlights the importance of best practices when creating the Text index and also the on-going maintenance.

Following sections are aimed at providing more information in this regard for Oracle Identity Manager administrators and database administrators.

- Section 13.4.5.1, "One-Time Optimizations for Oracle Text Index"
- Section 13.4.5.2, "Text Index Optimization"

13.4.5.1 One-Time Optimizations for Oracle Text Index

When you install Oracle Identity Manager, the Text index for Access Request Catalog is created with possible optimizations. However, Oracle Text has some more optimizations that are better applied based on the characteristics of the deployment. Following are the optimizations that you should consider applying for improving Access Request Catalog search performance. It is important to note that Access

Request Catalog is not usable when applying these and these are recommended to be done during a scheduled maintenance window.

> **Note:** Catalog Synchronization Job and Access Request Catalog should be down when these one-time optimizations are applied.

Storage of Text Index

Oracle Text index is stored in relational tables (DR$) which are presently resides in the default tablespace of Oracle Identity Manager schema. It is recommended to separate them out to their own tablespace. You can use the following commands to do that. You are recommended to be familiar with these steps and also make changes where needed.

1. Login to SYS schema and create a new tablespace to hold the text index internal tables. You can use the following sample command for it. 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 catalog_text_ind_tables
 DATAFILE 'DATA_DIR/catalog_text_ind_tables_01.dbf' SIZE 2048M REUSE
 EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

2. Connect to the database using Oracle Identity Manager schema.

3. Create a storage preference using the commands below. Oracle recommends you to be familiar with BASIC_STORAGE clause of Oracle Text and add more storage clauses if required. You can find more info on BASIC_STORAGE in Oracle Text Reference document.

```
Begin
Ctx_Ddl.Create_Preference('cat_storage', 'BASIC_STORAGE');
End;
/

Begin
ctx_ddl.set_attribute('cat_storage','I_TABLE_CLAUSE','tablespace
catalog_text_ind_tables storage (initial 5M next 5M)');
End;
/

Begin
ctx_ddl.set_attribute('cat_storage', 'K_TABLE_CLAUSE','tablespace
catalog_text_ind_tables storage (initial 5M next 5M)');
End;
/

Begin
ctx_ddl.set_attribute('cat_storage', 'R_TABLE_CLAUSE','tablespace
catalog_text_ind_tables storage (initial 1M) lob (data) store as (cache)');
End;
/

Begin
ctx_ddl.set_attribute('cat_storage', 'N_TABLE_CLAUSE','tablespace
catalog_text_ind_tables storage (initial 1M)');
End;
/

Begin
```

```
ctx_ddl.set_attribute('cat_storage', 'I_INDEX_CLAUSE','tablespace
catalog_text_ind_tables storage (initial 1M) compress 2');
End;
/
```

4. Apply the new storage preference using the following command. Make sure the Text index status is valid after this step.

```
ALTER INDEX CAT_TAGS rebuild parameters ('replace storage cat_storage');
```

5. Verify that the above tables are moved to the new tablespace by querying USER_SEGMENTS table.

KEEP Pool Settings for Text Index:

Oracle recommends put all the tables that make up the Text index in database KEEP pool to improve the performance of Access Request Catalog search. You must size the KEEP pool (DB_KEEP_CACHE_SIZE) correctly so that these Text index tables and other Oracle Identity Manager objects are retained in KEEP pool. To do so:

1. Connect to the database using Oracle Identity Manager schema.
2. Compute the size of the text index using the following query and use that to set/adjust DB_KEEP_CACHE_SIZE accordingly.

```
SELECT ctx_report.index_size('CAT_TAGS') FROM dual;
```

3. Run the following commands as Oracle Identity Manager schema user to put the tables in KEEP pool.

```
ALTER INDEX DR$CAT_TAGS$X STORAGE (buffer_pool keep);
ALTER TABLE DR$CAT_TAGS$R STORAGE (buffer_pool keep);
ALTER TABLE DR$CAT_TAGS$R STORAGE (buffer_pool keep) MODIFY lob (data)
(STORAGE (buffer_pool keep));
ALTER TABLE DR$CAT_TAGS$K STORAGE (buffer_pool keep);
ALTER TABLE DR$CAT_TAGS$I STORAGE (buffer_pool keep);
```

13.4.5.2 Text Index Optimization

The Text index could become fragmented due to on-going "Catalog Synchronization" Optimizing the text index on regular basis removes the old data and minimizes the fragmentations, which can improve the search performance of Access Request Catalog. To perform this, Oracle Identity Manager has introduced the following Oracle Database scheduler jobs:

- FAST_OPTIMIZE_CAT_TAGS
- REBUILD_OPTIMIZE_CAT_TAGS

These jobs reside in Oracle Identity Manager database schema and they are disabled by default. Oracle strongly recommends you to view these jobs, make schedule changes if needed and enable them. When changing the schedule, make sure the new schedule is set on the same line as the default schedule.

FAST_OPTIMIZE_CAT_TAGS meant to be running on frequent basis. By default, it is scheduled to run once a day at 1 AM. REBUILD_OPTIMIZE_CAT_TAGS does a full optimization and rebuilds the Text index. REBUILD_OPTIMIZE_CAT_TAGS is not meant to be running frequent basis. By default, REBUILD_OPTIMIZE_CAT_TAGS is scheduled to run every Sunday at 2 AM. Note that optimization may take a long time if your Text index is big.

Perform the following steps to change the schedule and/or enable these jobs.

1. Make sure the default schedule (daily 1 AM for FAST and every Sunday 2 AM for REBUILD) is acceptable to your environment. If not, change the schedule. If you are not sure, you can keep the default schedule and change later when needed.
2. Enable the jobs using the following commands:

```
BEGIN
DBMS_SCHEDULER.ENABLE ('FAST_OPTIMIZE_CAT_TAGS');
END;
/

BEGIN
DBMS_SCHEDULER.run_job ('REBUILD_OPTIMIZE_CAT_TAGS');
END;
/
```

Note: The Text index optimization can be done when the server is up and search of Access Request Catalog takes place.

13.5 Managing the Lifecycle of the Catalog

This section describes how to move Catalog customizations from a test environment to a production environment. You can extend the Catalog, customize the Catalog UI, and develop and test the customizations in a test environment, and then eventually roll out the customizations to your production environment.

This section includes the following topics

13.5.1 Overview of Catalog Customization

While the Access Request Catalog provides robust and rich out of the box functionality, there may be scenarios where you need to extend the Catalog and customize it to meet your business needs.

The following scenarios illustrate common scenarios where the Catalog may require customization.

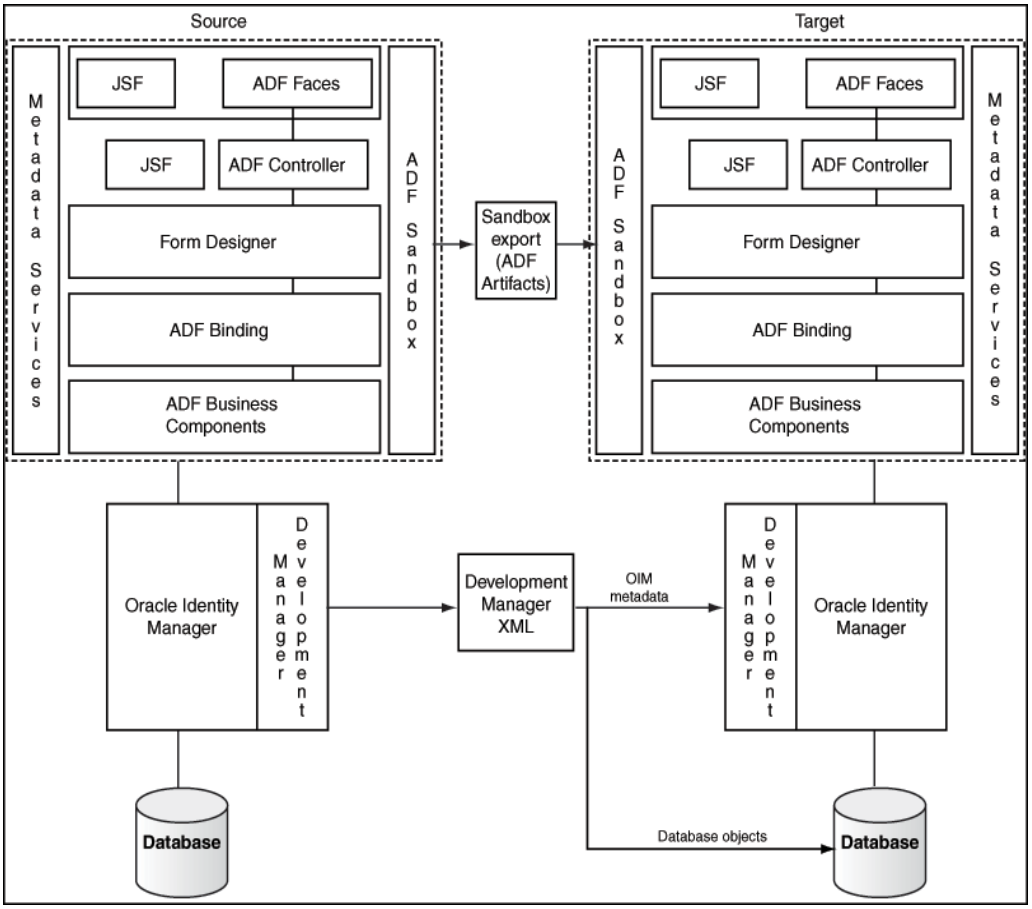
- MyCorp would like to add additional attributes, such as Cost to Line of Business and License Required, to give the requester an idea about the cost that would be incurred by the Line Of Business, when the requested item was granted. To support this scenario, the Catalog System Administrator extends the Catalog and adds two additional attributes, Cost to Line of Business and License required. Next, the administrator customizes the Catalog search results and Catalog item details page.

Note: In the request catalog, only String type of UDF can be created. If you mark that attribute as searchable attribute, it is of size 256 Char. If it is not a searchable attribute, then it is of size 2000 Char. You cannot mark a non-searchable attributes to searchable.

- MyCorp would like to show the Risk associated with an entitlement as part of Catalog search results. To support this scenario, the Catalog System Administrator customizes the Catalog search results and adds the item risk as an image widget.

These customizations are implemented by System Integrators or the customer's own IT staff and need to be moved to Test and to Production. Figure 13–2 shows the high-level process of moving customizations from Test to Production for the Catalog.

Figure 13–2 Test to Production Process for Catalog



Catalog customizations have three components:

1. ADF customizations

   ADF customizations include Catalog UI customizations including search results, item details, cart details and Catalog attributes added or modified using the Form Designer. These customizations should be done within a Sandbox session. For more information on Sandboxes, please refer to Section 13.5.2, "Test to Production procedures for Catalog customizations"

2. Oracle Identity Manager metadata customizations

   When you add new attributes to the Catalog entity or modify an existing attribute and change its properties, additional metadata is generated in Oracle Identity Manager. For example, if a new attribute, Secondary Approver, is added to the Catalog entity using the Catalog system entities, Oracle Identity Manager adds a database column corresponding to the attribute. If the attribute is searchable, Oracle Identity Manager stores additional metadata. These customizations should be moved from Test to Production using the Deployment Manager.

3. Data Migration

   The Catalog needs to be populated with relevant information, after adding/ modifying attributes in the Catalog to make the Catalog business-friendly and provide enough information so that users can use the Catalog effectively. Once this additional information, also referred to as the Glossary, has been reviewed and approved, it needs to be moved to Production.

13.5.2 Test to Production procedures for Catalog customizations

This section describes the steps to perform for moving the Catalog definition from Test to Production. It consists of the following steps:

- Section 13.5.2.1, "Exporting using the Sandbox and Deployment Manager"
- Section 13.5.2.2, "Importing Using the Deployment Manager and Sandbox"

Depending upon the type of customization done, you may need either one or both the steps. Use Table 13–2 to make a determination of which steps to carry out.

Table 13–2 Catalog Customization Steps

Customization	Sandbox required	Deployment Manager required
Adding/ Modifying a seeded Catalog attribute	Yes	Yes
Adding/ Modifying a Catalog UDF	Yes	Yes
Customizing Catalog UI	Yes	No
Populating Catalog	No	No

See Also:

- "Migrating Incrementally Using the Deployment Manager" on page 21-1 for detailed information about the Deployment Manager
- "Managing Sandboxes" in the *Developing and Customizing Applications for Oracle Identity Manager* for detailed information about sandboxes
- "Handling Concurrency Conflicts" in the *Developing and Customizing Applications for Oracle Identity Manager* for information about handling concurrency conflicts when multiple users customize an application by using sandboxes and troubleshooting concurrency issues

13.5.2.1 Exporting using the Sandbox and Deployment Manager

To Export Using Sandbox

To move the ADF customizations from Test to Production, follow the steps given below:

1. Login to Oracle Identity System Administration as a member of the System Administrator role.

Note: In scenarios where you need to switch between the Self Service (or Identity) and System Administration consoles and the Oracle Identity Manager 11g R2 deployment is not protected by Single Sign On, you must log out of one console before logging in into another.

2. Click **Sandbox** and select the Sandbox to be exported.
3. Click **Export Sandbox**. A sandbox can be exported as a file for transporting, sharing, and other usages where packaging it as a file is required.
4. Specify a file location for the zip file created.

To Export Using Deployment Manager

Note: Make sure that you do not have any popup blockers enabled in your browser and that you have a supported Java Runtime Environment (JRE) installed in the browser. This is because the Deployment Manager uses a popup window and it requires JRE to be installed in the browser.

To export the Oracle Identity Manager metadata from Test to Production, follow the steps given below:

1. Login to Oracle Identity System Administration as a member of the System Administrator or System Configurator role.
2. In the left pane, under System Configuration, click **Export**.
3. Select **Catalog Metadata** as the object to be exported.
4. Enter * in the search field and click **Search**.
5. Follow the steps to generate the Deployment Manager XML.

Note: Perform the following optional steps as a best practice:

- Backup/Check-in the sandbox zip file and the Deployment Manager XML as a single file into a source code control system like Subversion, SourceSafe, and so on.
- Repeat the steps above in the target (Production) environment and backup the Catalog entity and the Catalog UI.

13.5.2.2 Importing Using the Deployment Manager and Sandbox

Importing the customizations should be done in the reverse order. This is required since the ADF customizations expect the Oracle Identity Manager metadata to be present, when the ADF customizations are imported.

To Import Using Deployment Manager

To import the Oracle Identity Manager metadata from Test to Production:

1. Login to Oracle Identity System Administration as a member of the System Administrator or System Configurator role.
2. In the left pane, under System Configuration, click **Import**.

3. In the File browser popup, select the **Deployment Manager** XML file to be imported.
4. Follow the wizard steps to import the XML.

To import using the Sandbox

To move the ADF customizations from Test to Production:

1. Login to Oracle Identity System Administration as a member of the System Administrator role.

> **Note:** In scenarios where you need to switch between the Self Service (or Identity) and System Administration consoles and the Oracle Identity Manager 11g R2 deployment is not protected by Single Sign on, you must log out of one console before logging in into another.

2. Click **Sandbox** and then click **Import Sandbox.**
3. In the dialog, select the file to be imported.
4. In the left pane, under System Configuration, Click **Import**.
5. In the Sandbox Manager, select the sandbox and click **Publish Sandbox**.
6. Logout and log back in to view and verify the changes.

13.5.3 Limitations of the Test to Production procedures

There are some limitations in the Test to Production process for the Catalog, including the following:

- All ADF customizations must be done within a single sandbox session. While you can have multiple sandboxes, only one sandbox can be active at a time and as a result, changes in the System Administration Console i.e. Catalog entity extension and those done in the Identity Console, that is, Catalog UI customization, must be done in the same sandbox.
- Changes done outside a sandbox or done either before creating and activating a sandbox or after, are not visible in the sandbox.
- Once you publish a sandbox, you cannot export it or revert it. As a result, you must export the sandbox while it is still activated and not published and also ensure that you back your customizations before you import and publish a sandbox.
- Deployment Manager imports are committed immediately. There is no rollback capability in the Deployment Manager.

13.6 Troubleshooting

This section describes the troubleshooting procedures to be followed while resolving issues with the Access Request Catalog. It contains the following topics

13.6.1 Catalog synchronization issues

Catalog synchronization issues occur when roles, application instances and entitlements are not visible in the Access Request Catalog. Use the flow charts given below to troubleshoot synchronization issues for each of three Catalog item types that can be requested.

> **Note:** Harvesting job picks up the data for harvesting on the basis of the Update date parameter. If the update is blank, then all the records are fetched for processing.
>
> However, if the user has specified some date in the Update date parameter, only that data is processed which is created or updated after the given date.

- Troubleshooting synchronizing Roles with the Catalog

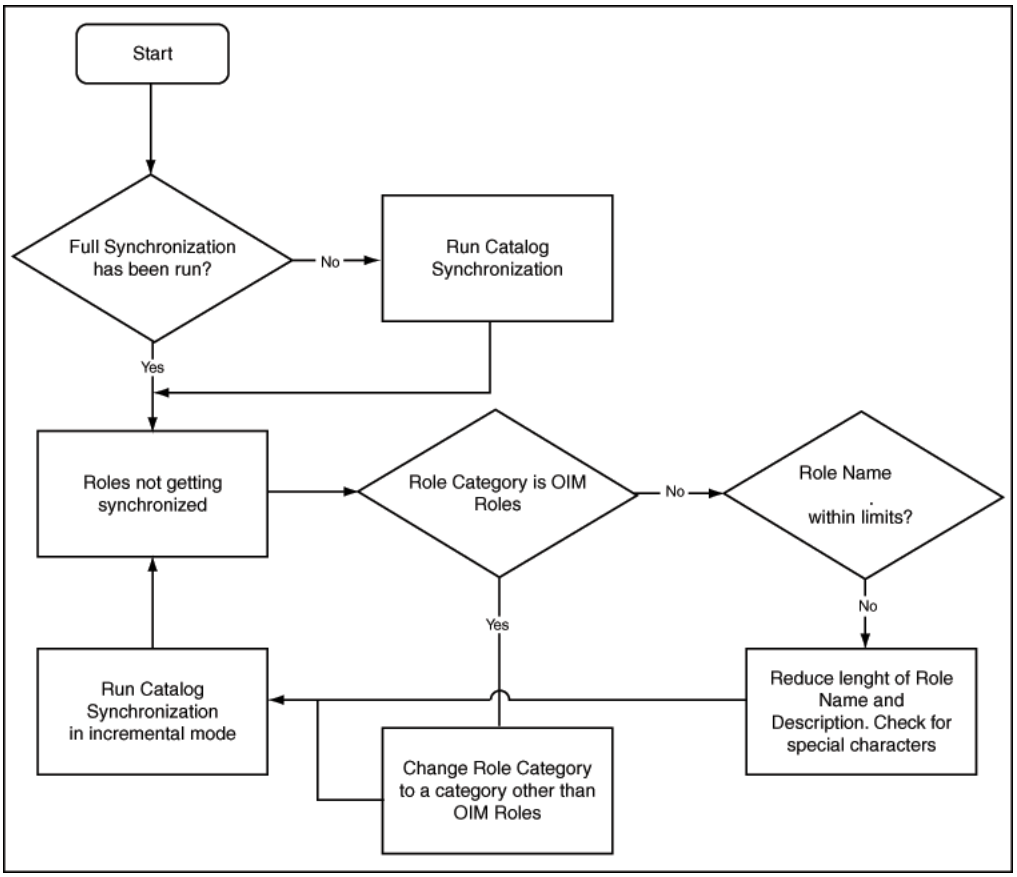
  The synchronization of Roles with the Catalog is real-time in nature. When a role is created, it is published to the Catalog immediately as long as it does not belong to the Oracle Identity Manager Roles category.

  > **Note:** The Oracle Identity Manager Roles role category is meant for Oracle Identity Manager usage only. Customers should not use this category for their enterprise Roles.

In a new Oracle Identity Manager 11g R2 installation, enterprise roles created by customers are available in the Catalog and the visibility is based on the organization scoping. In an upgraded environment, customers will have to run the Catalog Synchronization job in a bootstrap mode to publish the existing roles to the Catalog. New roles, created after upgrade, is available in the Catalog immediately.

Figure 13–3 shows a diagnostic flowchart that customers can use to troubleshoot scenarios where the roles created in Oracle Identity Manager are not visible in the Catalog.

Figure 13–3 Catalog Synchronization Diagnostic Flowchart

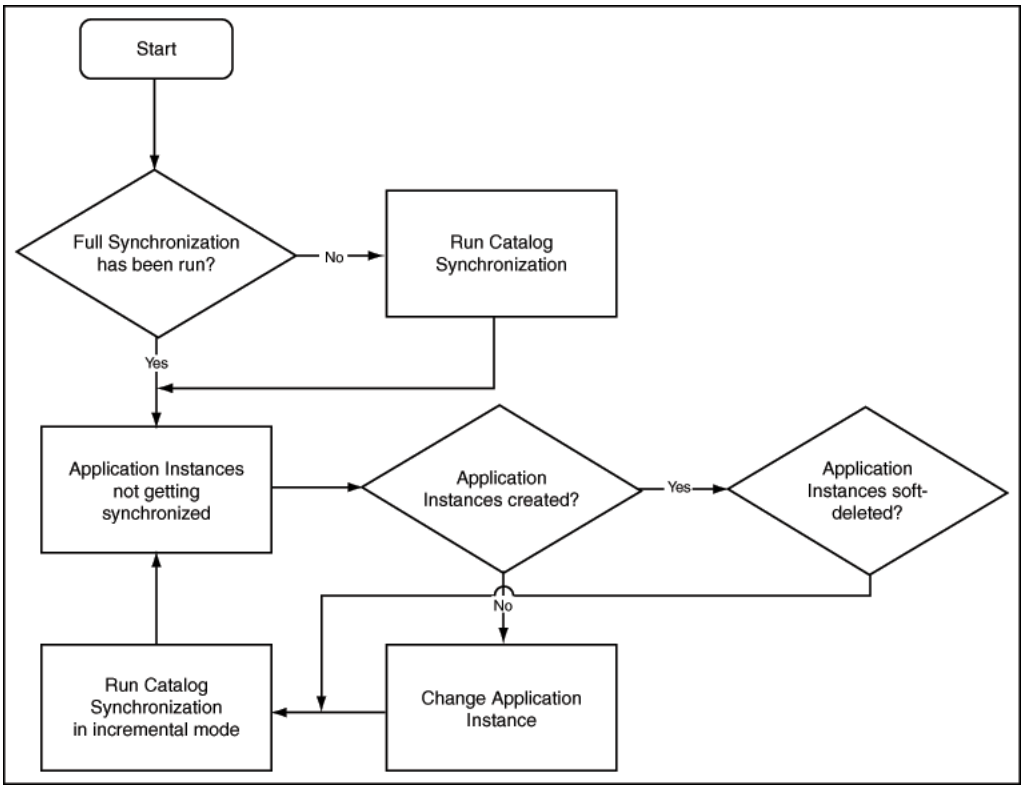


- Troubleshooting synchronizing Application Instances with the Catalog

  The synchronization of Application Instances with the Catalog is controlled by the Catalog Synchronization job. Application Instances require more configuration (than enterprise roles) and hence are not synchronized immediately with the Catalog.

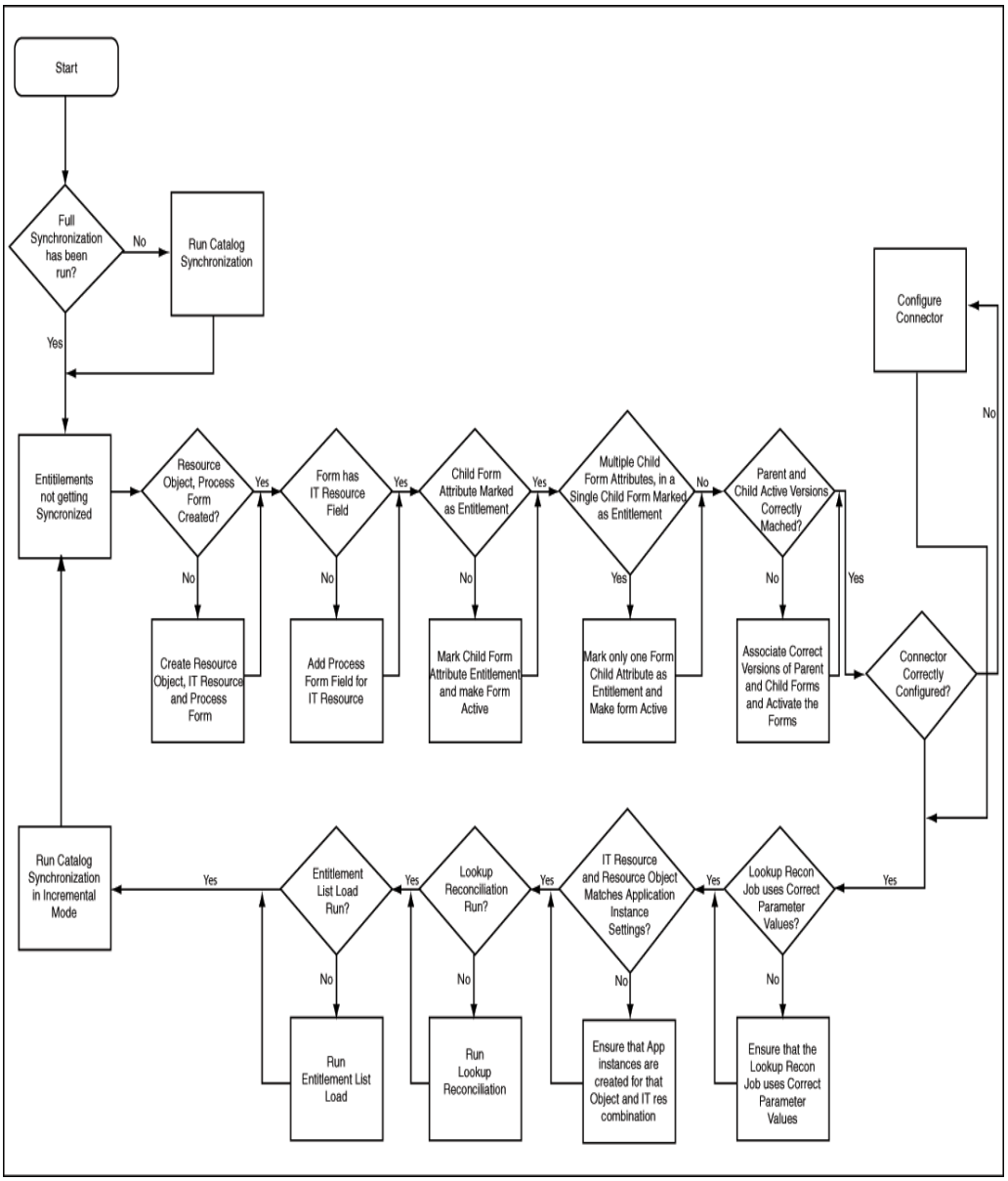
  Figure 13–4 shows a diagnostic flowchart to be followed when troubleshooting issues related to synchronizing application instances with the Catalog.

Figure 13–4 Trouble Shooting Synchronization Application Instances Flowchart



- Troubleshooting synchronizing Entitlements with the Catalog

Figure 13–5 Trouble Shooting Synchronizing Entitlements Flowchart



13.6.2 Catalog security issues

Catalog security is driven by two factors:

- The security model that uses Organization-based scoping for users, roles, application instances and entitlements. This security model controls what items a requester can see in the Catalog search results and the users who can be added as target users.
- The security model that is not scoped by organization and is used for global Admin Roles such as Catalog System Administrator.

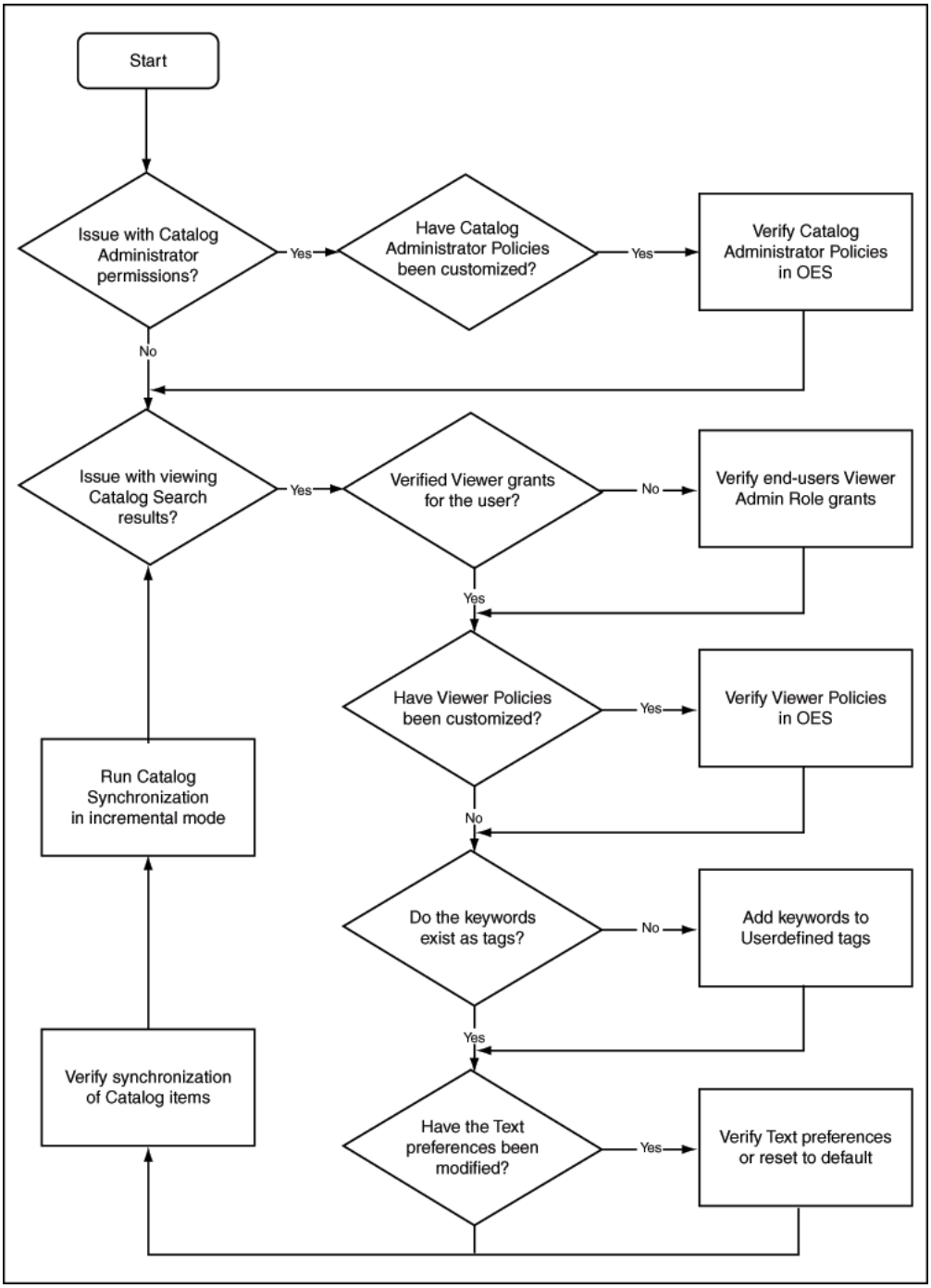
Typical issues with Catalog security are:

- Requesters cannot see the Catalog item even though they have entered the correct search keyword.

- Requesters are not able to add target users to the request
- Requesters are not able to provide additional information for application instance requests
- Requesters cannot see Catalog Item details such as Approver User, Approver Role, Fulfillment User, and Fulfillment Role.
- Catalog System Administrators do not see the Catalog Item in an edit mode and are not able to edit the Catalog Item
- Catalog System Administrators are not able to create Request Profiles

Figure 13–6 shows a diagnostic flow chart to be followed to troubleshoot issues with Catalog security.

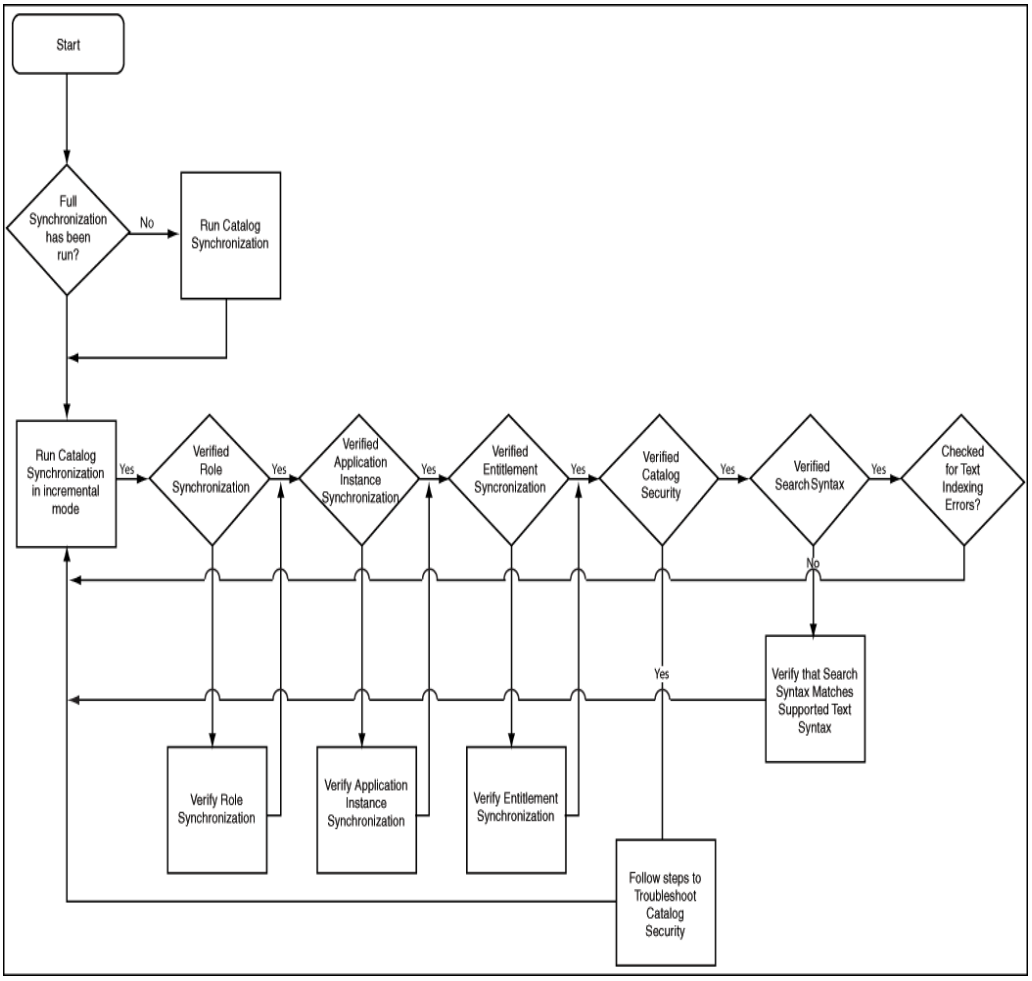
Figure 13–6 Diagnostic Flowchart With Security Issues



13.6.3 Catalog Search Issues

Figure 13–7 shows a diagnostic flow chart to be followed to troubleshoot issues with Catalog search.

Figure 13–7 Catalog Search



Note: Search Criteria for Catalog API's `findCatalog` method supports only AND conjunction operator.

13.6.4 Common Reasons for Request Failure

When the associated operations specified in a request fail to execute, the request cancels any pending operations and moves the request to the Request Failed stage. Clicking the Request Failed hyperlink displays the reason for request failure.

A request can fail for any one of the following reasons:

- If you are requesting a role, then your request can fail due to an SoD violation.
- If you are requesting an application instance and that application instance depends on another application instance, then the request moves to 'Request Approved Fulfillment Pending' status because the parent application instance is not provisioned. For example, to successfully provision a user to a Microsoft Exchange account, the user must have a Microsoft Active Directory account in the domain controller that is managing the users of the Exchange server.

In addition to the preceding reasons, failures can occur because of incorrect password, password policy violation, target system being unavailable, and so on.

Part VII

System Configuration

This past describes system configuration in Oracle Identity Manager.

It contains the following chapters:

14

Managing Home Organization Policy

When an user submits a request for self-registration, the home organization of the user gets determined by the home organization policy. The organization name, as determined by the home organization policy, is filled in the request submitted. The approver can override the home organization of the user while approving the request. If a pre-process custom handler is defined to determine the home organization during self-registration, then home organization policy is not evaluated. If workflow policy is defined, then it takes precedence over the home organization policy.

In home organization policy, you can define rules based on user attributes. The return value of the rule is the organization name. Rules are evaluated in the order in which they appear on the Home Organization Policy page in Oracle Identity System Administration, starting from first rule to the last rule. Rules can be re-ordered from the Home Organization Policy page. Rule evaluation stops when a rule matches and the organization name is returned. The remaining rules are not evaluated.

This chapter includes the following sections:

14.1 Features of Home Organization Policy

During Oracle Identity Manager deployment, a default home organization policy called **Home Organization Determination Policy** and a default rule called **Default All Users To Single Organization** is seeded, if not already present. Oracle Identity Manager does not allow you to define new home organization policies. However, new rules can be created under the default home organization policy.

The Default All Users To Single Organization rule is satisfied by every user. If for any reason the default rule is deleted, then if a user does not satisfy any other rule, then home organization of that user is left blank in the request submitted. The approver can fill in the home organization name before approving. When SOA server is disabled, approver cannot fill in the home organization name, hence blank home organization field will result in request failure. Ensure that rules are defined in such a way that every user will satisfy at least one rule and a home organization is assigned.

Rules in home organization policy can be defined using Text, Number, Checkbox and Date Type UDFs. However, LookUp Type UDFs cannot be added to the

self-registration page. List of operators available to build the IF condition is different for each type of UDF.

Following use cases shows how Home Organization Policy works:

- Self Registration Use Case Using Default Rule
- Self Registration Use Case Using Simple Rule
- Self Registration Use Case Using Complex Rule
- Rule Evaluation Order
- Self Registration Use Case When SOA is OFF

14.1.1 Self Registration Use Case Using Default Rule

Default rule is named as **Default All Users To Single Organization Rule**. This rule can be modified but cannot be deleted.

The condition defined is:

```
IF user.User Login  Equals  $(user.User Login) THEN organization equals "Xellerate Users"
```

The default condition always evaluates to True. Thus if any other rule defined in Home Organization Policy does not get satisfied, the default rule will definitely be satisfied and will provide the home organization name.

For example, when an user with userLogin **User1** submits a self registration request, and if no other rule is defined or satisfied, default rule is evaluated. And the home organization is set to **Xellerate Users**.

14.1.2 Self Registration Use Case Using Simple Rule

A simple rule is a rule created with a single IF condition and with out using any operator like AND/OR.

For example, if a rule called **ExampleSimpleRule** is defined with the following condition:

```
IF user.Nickname Starts with "Test" THEN organization equals "testOrg2"
```

Here, user.Nickname is a text UDF attribute.

Now if a user with nickname as **TestUser2** submits a self-registration request, then the rule condition is satisfied and home organization is set to **testOrg2.**

14.1.3 Self Registration Use Case Using Complex Rule

A complex rule is a rule created with more than one IF condition and uses AND/OR operators to form the rule.

For example, if a rule called **ExampleComplexRule** is defined with the following condition:

```
IF user.Nickname Starts with "Test" AND user.Display Name Ends with "User" THEN organization equals "testOrg3"
```

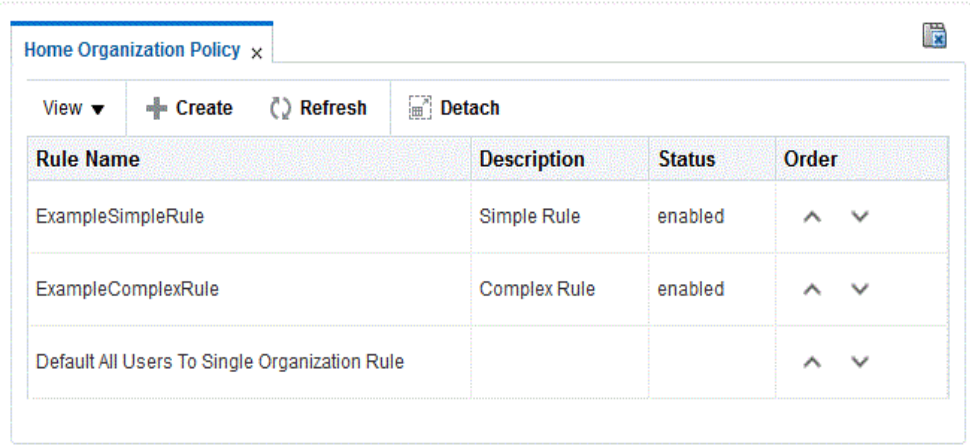
Here, user.Nickname is a UDF attribute and user.Display Name is default attribute.

Now if a user with nickname as **TestUser3** and display name as **testUser** submits a self-registration request, then the rule condition is satisfied and home organization is set to **testOrg3**.

14.1.4 Rule Evaluation Order

When a user self registers, the first rule that is evaluated is the top rule on the list that appears on the home organization page, followed by the next rule up to the last rule. Evaluation stops as soon as a match is found. For example, if the **ExampleSimpleRule** is created followed by **ExampleComplexRule** as shown in Figure 14–1.

Figure 14–1 List of Rules Defined in Home Organization Policy Page



Then when a user self registers, user attribute values are evaluated against **ExampleComplexRule** first, if it does not match, it proceeds to evaluate against **ExampleSimpleRule**. If this also does not match it is evaluated against **Default All Users To Single Organization Rule** which is the default rule.

If evaluation against **ExampleSimpleRule** is satisfied, then home organization of the user is set according to the condition in the rule.

14.1.5 Self Registration Use Case When SOA is OFF

When SOA is off, and a self registration request is submitted, then the request gets auto-approved and status of request is shown as completed.

For steps to disable SOA server refer to "Disabling SOA Server" on page 4-31.

Now when a user submits a self registration, the status is shown as complete as the request is auto-approved. Evaluation of home organization rule is same as explained in the examples above.

14.2 Creating a Rule in Home Organization Policy

To create a rule in home organization policy:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Home Organization Policy**. The Home Organization Policy window is displayed.
3. Click **Create** on the toolbar. The Add Home Org Policy Rule page is displayed.

4. Under the Create Rule section, enter **Name**, **Description**, **Owner**, and **Status** for the new rule. Status of a rule can be set to Enable or Disable. If the Status is set to Disable, then when a user self registers, this rule is skipped during evaluation.
5. Set the rule condition in Condition Builder section. For example, If Display name contains Test and Last name contains User, then Organization is Vision North America. In this example Attribute is Display name, condition is contains and value is test.

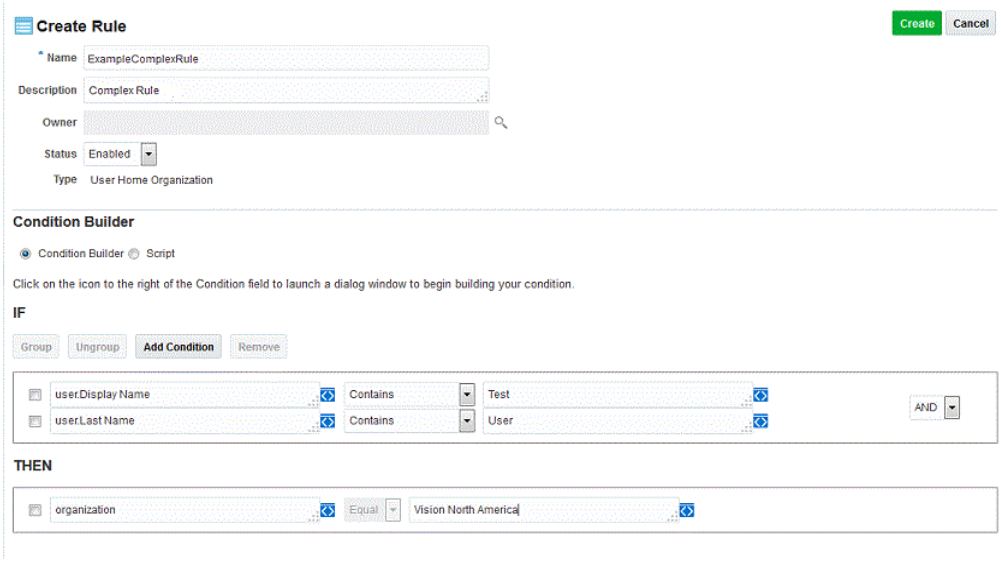
   You can set the rule using Condition Builder or Script.

   - To set rule using Condition Builder do the following:

     a. Under IF part of the rule, to enter attribute, click the Condition Builder icon. Condition builder pop-up screen is displayed.

     As an example, Figure 14–2 shows the Create Rule page with Condition Builder option to set rule.

Figure 14–2 Creating Rule With Condition Builder Option



b. Select the User attribute for the attribute list, list of UDF and default attribute associated with User is listed.

Search for the particular attribute from the list or type the name of the attribute in the text box and click the **Search** icon. Select the attribute form the list and click **OK**.

c. Select the condition from the conditions drop-down. The available conditions are, Equal, Not Equal, Contains, Does Not Contain, Begins With, Does Not Begins With, Ends With, and Does Not Ends With.

Note: This list varies based on the type of attribute. The list above is for text type. Number type attributes can have values Greater than, Lesser than and so on.

---

d. To enter value, type the value in the text box and click **OK** or click the **Value** icon to open the condition builder pop-up screen.

In the condition builder, you can opt to enter **Value** or **Expression**.

If you select **Value**, list of value is displayed. Select the required value or type the value in the text box and click **OK**.

If you select **Expression**, list of condition is displayed. Select the required value and click **OK**.

e. To enter the THEN part of the rule, click the organization icon. Condition builder pop-up screen is displayed. Select organization and click **OK**.

f. Condition is by default set to Equals and cannot be changed.

g. To select the organization, click the organization name icon. Condition builder pop-up screen is displayed. Select the organization name from the list and click **OK**.

- Support for groovy expressions is provided by default, for which a script can be used.

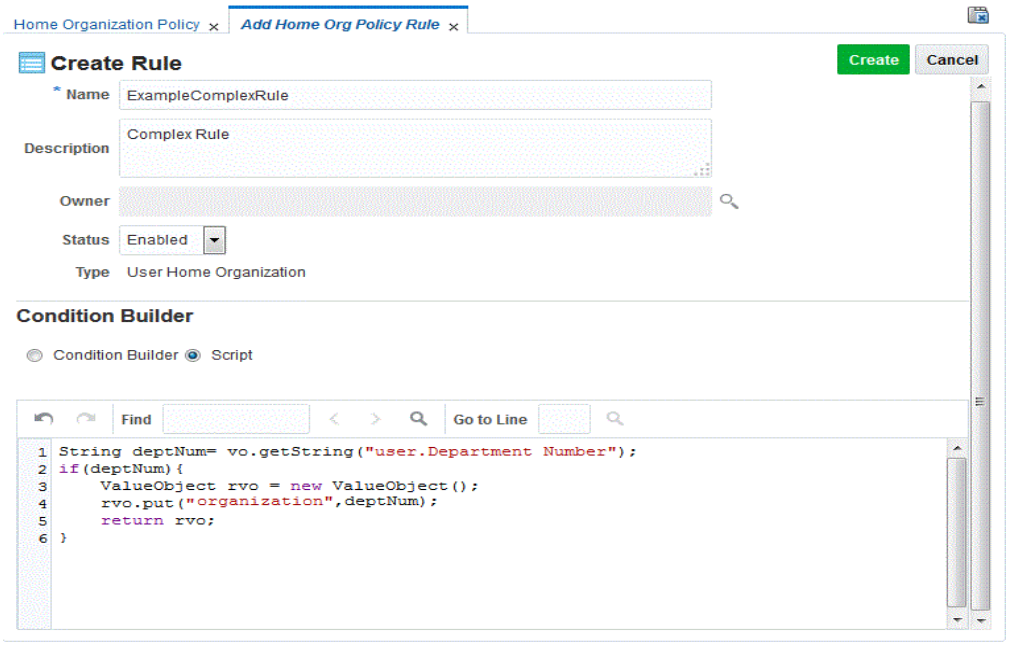
  To set rule using a script, perform the following:

  a. When **Script** is selected, this section shows the existing script. For example, if user has department number configured, then set organization value as department number. If department number is Oracle, Oracle-HQ, or Oracle-IDC, then organization value is set to department number. Make sure that organization with name Oracle, Oracle-IDC, Oracle-HQ exists in the system.

```
String deptNum= vo.getString("user.Department Number");
if(deptNum)
{
ValueObject rvo = new ValueObject();
rvo.put("organization",deptNum);
return rvo;
}
```

  As an example, Figure 14–3 shows the Create Rule page with **Script** option to set rule.

Figure 14–3 Creating Rule With Script Option



b. Enter any word you would want to find and click the **Search** icon. Find and Replace panel is displayed.

c. To jump to a particular line, enter line number and click the **Search** icon.

6. To set complex rules click **Add Condition**. Select **AND** or **OR** condition and set additional rule by following instruction in Step 5.
7. Click **Create**.
8. The Home Organization Policies page lists all the rules defined. The defined rule can be moved up or down in the list to change its order, to do so click the Up or Down arrow in the Order column of the rule.

14.3 Modifying a Rule in Home Organization Policy

To modify a rule in home organization policies:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Home Organization Policy**. The Home Organization Policy window is displayed.
3. Select the required home organization policy from the list and click **Open**.
4. Modify the required details and click **Update**.

   If you do not wish to update the changes made to the rule, click **Revert**. The rule is restored to the original rule.

14.4 Deleting a Rule in Home Organization Policy

To delete a rule in home organization policy:

1. Login to Oracle Identity System Administration.

2. In the left pane, under System Configuration, click **Home Organization Policy**. The Home Organization Policy window is displayed.
3. Select the home organization policy that needs to be deleted from the list and click **Delete**.

15

Managing Self Service Capability Policy

Oracle Identity Manager allows you to control what operations a user can perform for the self. For example, if a user belongs to a particular organization, then the user is allowed only to change self profile, and other operations in Oracle Identity Manager are restricted. This can be achieved by setting rules in the Self Service Capability Policy. In Self Service Capability Policy, you can define rules based on user attributes. You can set user attributes as denied attributes for the user who satisfies the rule. The user attributes marked as denied attributes cannot be viewed or edited. The return value of this rule is the capability assigned to the user and the denied attributes that are configured. Self Service Capability Policy is seeded with a default rule.

Multiple self service capability rules can be configured. The evaluation of these rules is based on their order. The order can be configured from the Self Service Capability page in Oracle Identity System Administration. All the rules are evaluated one by one and capabilities of the first matching rule are assigned to the user.

This chapter includes the following sections:

15.1 Default Self Service Capability Rule

The Self Service Capability Policy is seeded with a **Default Self Service Capability** rule. The default condition always evaluates to true. Therefore, if any other rule defined in the Self Service Capability Policy is not satisfied, the default rule is satisfied and provides the user with all the self service capabilities.

15.2 Example of Self Service Capability Rules and Rule Evaluation Order

Example of rules that can be set are:

- If user type is Contractor, then user is allowed only to manage self profile.

  ```
  If user.Role Equal Contractor THEN capability Equal selfModifyUser
  ```

- If user type is Full Time and belongs to Sales department, then user is allowed to request roles and modify their profiles.

```
If user.Role Equal Full-time AND user.Department Number Equal Sales
THEN
capability Equal addSelfRoles
AND
capability Equal selfModifyUser
```

- If user type is Full Time and country is not USA, then user is allowed to modify their profiles and Middle Name is a denied attribute to this user.

```
If user.Role Equal Full-time AND user.Country Not Equal USA
THEN
capability Equal selfModifyUser
AND
deniedAttribute Equal Middle Name
```

- If user type is Full Time and country is USA, then user is allowed to modify their profiles.

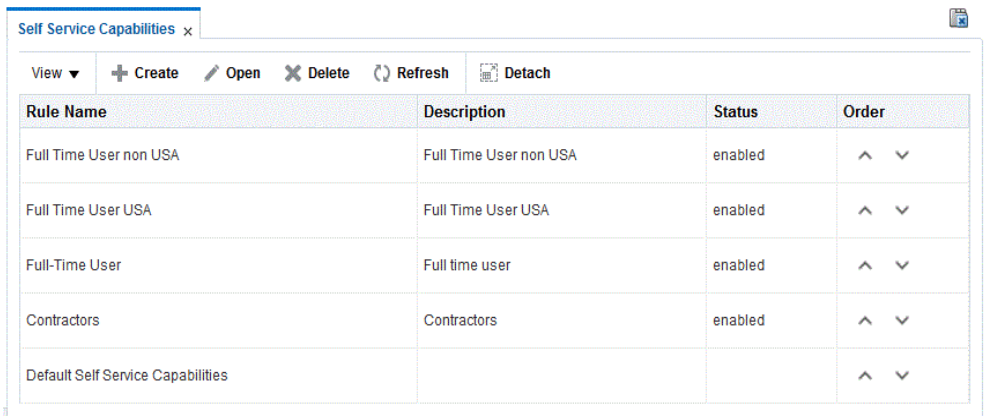
```
If user.Role Equal Full-time AND user.Country Equal USA
THEN
capability Equal selfModifyUser
```

When a user is created, the first rule that is evaluated is the latest defined rule, followed by the next latest up to the default rule. Evaluation stops as soon as a match is found.

For example, consider that, **Contractor** rule is created first, followed by **Full-Time User**, **Full Time User USA**, and **Full Time User non USA**. Figure 15–1 shows the order of rules.

When a user is created, user attribute values are evaluated against **Full Time User non USA** first, if it does not match, it proceeds to evaluate against **Full Time User USA**. If this does not match it is evaluated against **Full-Time User** and then **Contractor**. If non of these rules match, then it is evaluated against the default rule, that is **Default Self Service Capability**. If evaluation against **Full Time User non USA** is satisfied, then capability of the user is set according to the condition in the rule.

Figure 15–1 List of Rules Defined in Self Service Capabilities Page



The order of the rule can be modified using the arrow buttons in the **Order** column of the rule.

15.3 Creating a Rule in Self Service Capability Policy

To create a rule in self service capabilities:

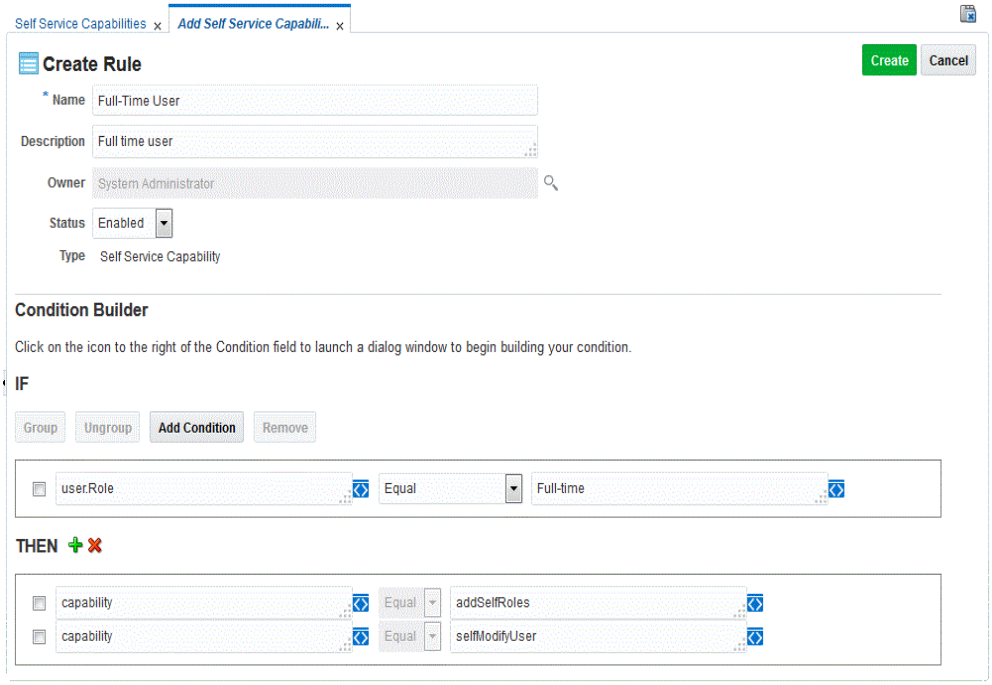
1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Self Service Capabilities**. The Self Service Capabilities page is displayed.
3. Click **Create** on the toolbar. The Add Self Service Capability Policy Rule page is displayed.
4. Under the Create Rule section, enter **Name**, **Description**, **Owner**, and **Status** for the new rule. Status of a rule can be set to Enable or Disable. If the Status is set to Disable, then when a user is created, this rule is skipped during evaluation.
5. Set the rule condition in the Condition Builder section. For example,

   To set rule using Condition Builder:

   a. Under IF part of the rule, to enter attribute, click the condition builder icon. Condition builder pop-up screen is displayed.

      As an example, Figure 15–2 shows the Add Rule page.

Figure 15–2 Creating Rule With Condition Builder Option



   b. Select the User attribute from the attribute list. List of searchable attributes and UDFs associated with User are listed.

      Search for the particular attribute from the list or type the name of the attribute in the text box and click the **Search** icon. Select the attribute from the list and click **OK**.

   c. Select the condition from the conditions drop-down. The available conditions are, Equal, Not Equal, Contains, Does Not Contain, Begins With, Does Not Begins With, Ends With, and Does Not Ends With.

Note: This list varies based on the type of attribute. The list above is for text type. Number type attributes can have values Greater than, Lesser than and so on.

d. To enter value, type the value in the text box and click **OK** or click the **Value** icon to open the Condition builder pop-up screen.

In the condition builder, you can opt to enter Value or Expression.

If you select Value, list of value is displayed. Select the required value or type the value in the text box and click **OK**.

If you select **Expression**, list of condition is displayed. Select the required value and click **OK**.

Note: This field is case sensitive.

e. To enter the THEN part of the rule, click the condition builder icon. Condition builder pop-up screen is displayed. Select **Capability** or **Denied Attributes** and click **OK**.

f. Condition is set to **Equals** and cannot be changed.

g. To select the Capability or Denied Attribute based on the selection in previous step, click condition builder icon under THEN section. Condition builder pop-up screen is displayed. Select the desired default capability or denied attribute from the list and click **OK**.

Note:

- Mandatory attributes and System generated attributes like Status, Display name, User Login and so on cannot be included in denied attributes list.
- When denied attributes are specified, the user will not be able to view or modify those attributes.

6. To set complex rules click **Add Condition**. Select **AND** or **OR** condition and set additional rule by following instruction in Step 5.
7. Click **Create**.

15.4 Modifying a Rule in Self Service Capability Policy

To modify a rule in self service capabilities:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Self Service Capabilities**. The Self Service Capabilities window is displayed.
3. Select the self service capability you want to modify from the list and click **Open**.
4. Modify the required details and click **Update**.

   If you do not wish to update the changes made to the rule, click **Revert**. The rule is restored to the original rule.

15.5 Deleting a Rule in Self Service Capability Policy

To delete a rule in self service capabilities:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Self Service Capabilities**. The Self Service Capabilities window is displayed.
3. Select the self service capability that needs to be deleted from the list and click **Delete**.

16

Managing Lookups

This chapter describes how to manage lookups in Oracle Identity Manager by using the Form Designer in the Oracle Identity System Administration.

> **Note:** Oracle Identity Manager does not support lookup queries.

The Form Designer in the Oracle Identity System Administration enables you to perform the following:

- Searching a Lookup Type
- Creating a Lookup Type
- Modifying a Lookup Type

16.1 Searching a Lookup Type

To search for a lookup type:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Lookups**. The Search and Select: Lookup Type window is displayed.
3. 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.
4. In the Meaning field, enter the humanly readable description of the lookup value you want to search.

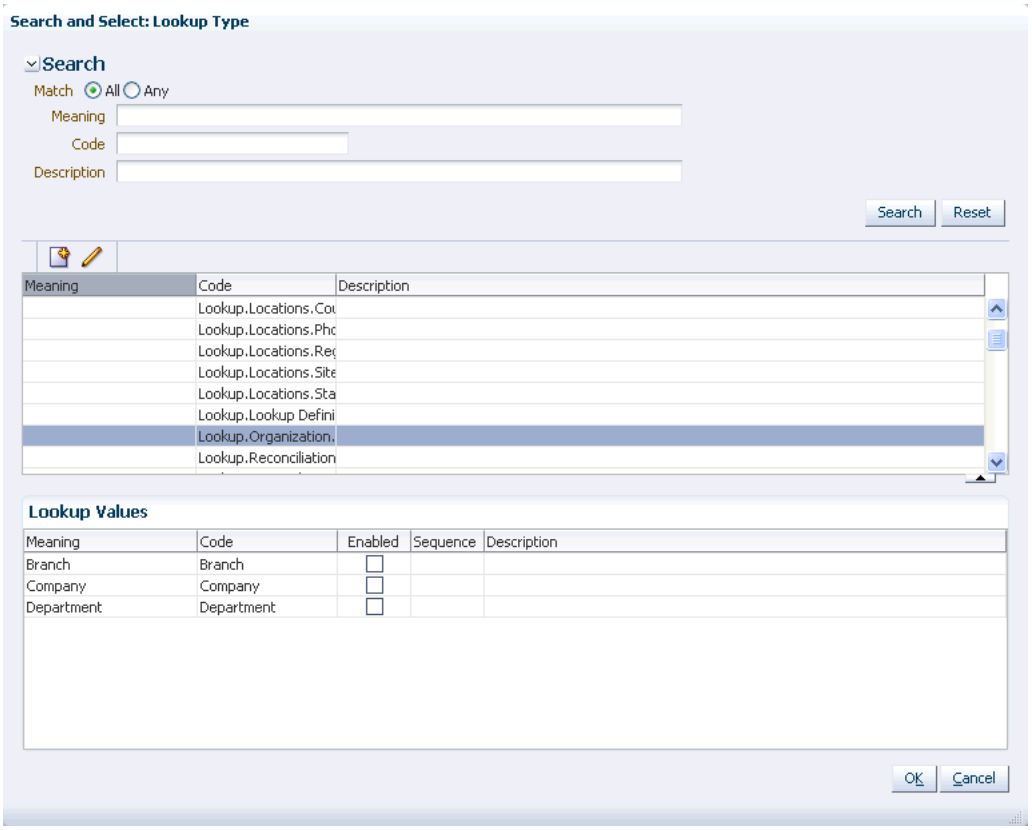
> **Note:** Meaning is the decoded value, and Code is the encoded value. The value in the Meaning field is a humanly readable description of the field. The value in the Code field is the actual code value that is used for provisioning. For example, decoded value can be a LDAP group name, and encoded value is the LDAP group GUID.

5. In the Code value, enter the Code value of the lookup type that you want to search.

> **Note:** To specify the search criteria, you can use the percent (%) wildcard character.

6. In the Description field, you can enter a description of the lookup type.
7. Click **Search**. The lookup types that match your search criteria are displayed in a tabular format.
8. Select a row in the search results table. The details of the selected lookup type is displayed in the Lookup Values section, as shown in Figure 16–1:

Figure 16–1 The Search and Select: Lookup Type Window



9. The lookup values are enabled by default. You can deselect the checkboxes in the Enabled column for each lookup value to disable the lookup value.

Note: There are multiple ways in which lookups are used. One way is to populate some form with data via the lookup icon on some process form to provision to a target system. Many lookups, such as lookups for most connectors, contain some configuration information. These lookups do not honor the checkbox in the Disabled column and assume that all configuration settings are valid.

Task triggering based on lookup.usr_process_triggers, does not take into account or depend upon enabling and disabling of lookup value. If an entry is made into the lookup and the corresponding task is defined, then the task is triggered.

To workaround this, either change the task name at process definition or change the value in the lookup definition level for task name. Oracle recommends changing the value in the lookup definition level for task name as a better approach.

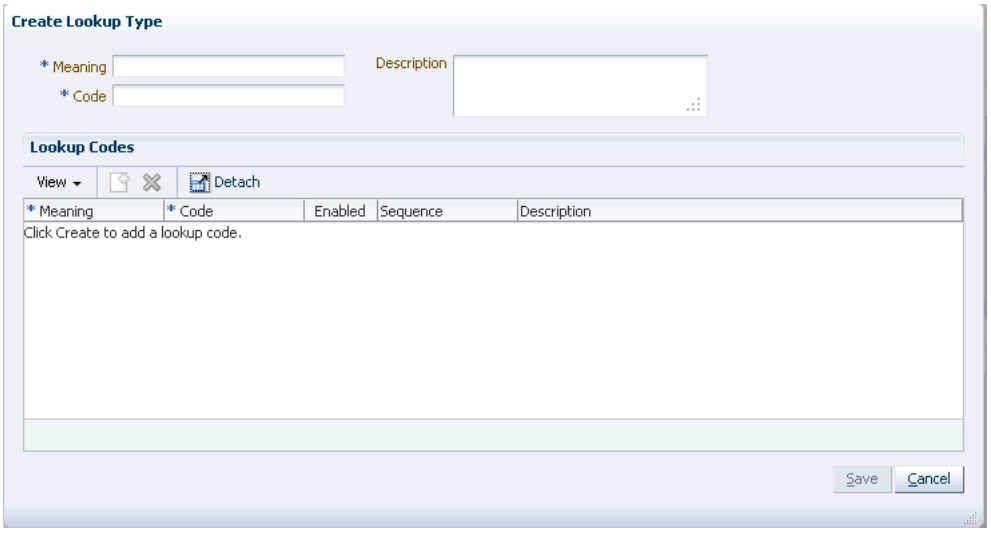
10. When finished, click **OK**.

16.2 Creating a Lookup Type

To create a lookup type:

1. Open the Search and Select: Lookup Type window.
2. Click the Create Lookup Type icon on the toolbar. The Create Lookup Type dialog box is displayed, as shown in Figure 16–2:

Figure 16–2 The Create Lookup Type Dialog Box



3. Enter values in the Meaning and Code fields. These are mandatory fields. For a description of the Meaning and Code fields, see step 4 in "Searching a Lookup Type" on page 16-1.
4. In the Description field, optionally enter a description of the lookup type.
5. Create one or more lookup codes for the lookup type. To do so:

a. In the Lookup Codes section, click the Create Lookup Code icon. A row is added to the Lookup Codes section in which you can specify values for the attributes of the lookup code.

b. Enter values for the Meaning, Code, and Description attributes.

c. Select the checkbox in the Enabled column if you want to enable the lookup code.

d. Repeat steps a to c to create as many lookup codes you want. To remove a lookup code, you can select the row for the code, and click the Remove Lookup Code icon.

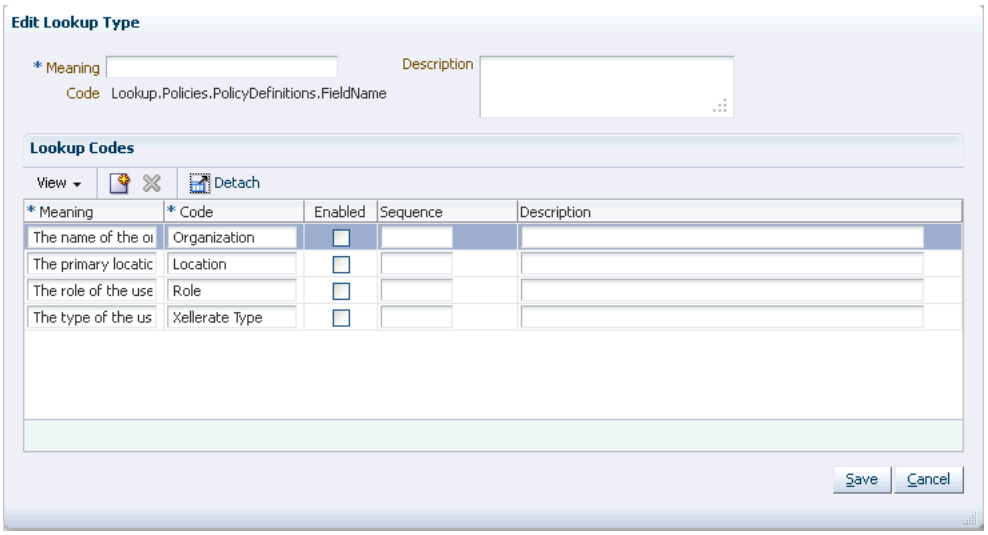
6. Click **Save**. The lookup type is created.

16.3 Modifying a Lookup Type

To modify a lookup type:

1. Open the Search and Select: Lookup Type window.
2. Click the Edit Lookup Type icon on the toolbar. The Edit Lookup Type dialog box is displayed, as shown in Figure 16–3:

Figure 16–3 The Edit Lookup Type Dialog Box



3. To modify the values of the Meaning and Description attributes, specify values in the respective fields. The Code field is a read-only field.
4. To modify lookup codes, select a row for the lookup code, and change the attribute values.
5. Add or remove lookup values by clicking the Create Lookup Code and Remove Lookup Code icons respectively. For more information, see step 5 of "Creating a Lookup Type" on page 16-3.
6. Click **Save**. The Lookup Type is modified.

Note: PurgeCache utility must be run after updating lookup definition, without which you must re-save lookup UDF in a sandbox before the new lookup values can be used. This is also applicable to predefined fields and their lookup definitions. Therefore, PurgeCache utility must be run to purge cache for all categories.

See *Oracle Fusion Middleware Performance and Tuning Guide* for information about purging the cache.

17

Managing Role Categories

Role categories exist in this release of Oracle Identity Manager only for the purpose of backward compatibility. Using role categories is not recommended.

If you are using a fresh deployment of Oracle Identity Manager, then use the Category attribute in the access catalog. If you are using an upgraded deployment of Oracle Identity Manager, then update the Catalog category attribute with the role category.

The default role categories in Oracle Identity Manager are:

- **OIM Roles:** All the predefined roles in Oracle Identity Manager are assigned to this category. These are roles that exist in Oracle Identity Manager by default and are primarily used for managing permissions. There will not be any corresponding entity in catalog for these predefined roles.
- **Default:** A newly created role must have a role category. Therefore, if a role category is not specified at the time of creating the role, then the role is assigned to this category by default.

Note: The default role categories cannot be localized.

This section describes the following topics:

17.1 Creating a Role Category

To create a role category:

1. Login to Oracle Identity System Administration.
2. On the left pane, under System Configuration, click **Role Categories.** The Search Role Categories page is displayed.
3. From the Actions menu, select **Create**. Alternatively, click **Create** on the toolbar. The Create Role Category page is displayed.
4. In the Role Category box, enter the name of the role category. This is a mandatory field.

5. In the Role Category Description box, enter a description for the role category. This step is optional.
6. Click **Save**. The role category is created, and the role category details page is displayed. This page consists of the Attributes tab.

   The Attributes tab displays the attributes of the role category. You can edit the fields in this tab to edit the role category.

17.2 Searching Role Categories

To search for role categories:

1. In Oracle Identity System Administration, Under System Configuration, click **Role Categories**. The Search Role Categories page is displayed.
2. In the Role Category field, specify a value. You can include wildcard characters (*) in the attribute value.
3. For the attribute value that you specify, select a search operator from the list. The following search operators are available:
   - Starts with
   - Ends with
   - Equals
   - Does not equal
   - Contains
   - Does not contain

   The search operator can be combined with wildcard characters to specify a search condition. The asterisk (*) character is used as a wildcard character. For example, you can specify the value to be D* as the search criteria, and select Equals as the search operator. The role categories that begins with D are displayed.
4. To add a searchable attribute to the Role Categories, click **Add Fields**, and select the attribute from the list of attributes.
5. If you want to change the order of the search fields, then click Reorder. The Reorder Search Fields dialog box is displayed. Move the search fields up or down by using the up and down arrows. When finished, click **OK**.
6. If you want to save the search criteria for future user, then click **Save**. See *Performing Self Service Tasks with Oracle Identity Manager* for information about saved search.
7. Optionally click **Reset** to reset the values that you specified as search conditions. Typically, you perform this step to remove the specified search conditions and specify a new search condition.
8. Click **Search**. The search result is displayed in a tabular format.

17.3 Modifying a Role Category

To modify a role category:

1. In the Search Role Categories page, search and select the role category you want to modify.

2. From the Actions menu, select **Open**. Alternatively, click **Open** on the toolbar. A page with details about the role category is displayed.

   You can also open the role category details by clicking the role category name.

3. The Attributes tab is open by default. Edit the fields in this tab to modify basic category information such as name and description. When finished, click **Apply**.

17.4 Deleting a Role Category

To delete a role category:

1. In the Search Role Categories page, search and select the role category you want to delete.
2. From the Actions menu, select **Delete**. Alternatively, click **Delete** on the toolbar.

   If the role category detail page is open, then click **Delete** on the toolbar.

   A message box is displayed asking for confirmation.
3. Click **Delete**. The role category is deleted. Alternatively, you can also delete the role category from its details page.

18

Managing the Scheduler

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 email 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 email to each employee manually. Alternatively, the system administrator can use a service, such as scheduler. In Oracle Identity Manager, there is a predefined scheduled task called Password Warning Task. The system administrator can use this scheduled task to create a scheduled job with the intended schedule.

> **See Also:** Table 18–2, " Predefined Scheduled Tasks" for information about the Password Warning Task scheduled task

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

A **scheduled task** configures the metadata for a job, which is to be run, and the parameters required for execution of that task. This metadata is predefined 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:

18.1 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 MDS. To access the oim-config.xml file by using Oracle Enterprise Manager:

1. Log in to Oracle Enterprise Manager.
2. Click **Application Deployments**.
3. Right-click **OIMAppMetadata(11.1.2.0.0)(*oim_server_name*)**, and select **System MBean Browser**.
4. In the System MBean Browser, navigate to **Application Defined MBeans**, **oracle.iam**, **Server: oim server**, **Application: oim**, **XMLConfig**, **Config**, **XMLConfig.SchedulerConfig**, **Scheduler**.

Table 18–1 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 18–1 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

Table 18–1 (Cont.) Child Elements of the Scheduler Element

Element Within Scheduler Element	Description
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

18.2 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 are stopped. When the scheduler gets the stopped status while jobs are running, the currently running jobs are stopped. In addition, the jobs that are scheduled to run does not run, but are submitted for run according to the schedule. When the Scheduler Service is up in the future, all submitted jobs are run.

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 Identity Manager server, and *OIM_PORT* refers to the port on which Oracle Identity Manager server is listening.

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

   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 stops the scheduler and further execution of triggers, but it does not stop or abort any jobs that are already executing. When the Scheduler Service is started again, jobs will then be executed at their appropriate times based on when they are scheduled.
   - **REINIT:** Click this button to reinitialize the scheduler.

18.2.1 Controlling Scheduler Start or Stop in a Clustered Environment

The scheduler.disabled system property is required if you want to control scheduler start or stop on a clustered setup. The scheduler.disabled system property must be set to true if you do not want to start the scheduler service on that node of the cluster.

This section contains the following topics:

18.2.1.1 Adding the Server Side Property for Oracle Identity Manager

To add the scheduler.disabled server-level property:

1. Log in to the WebLogic Administrative Console.
2. On the left panel, select **Environment**, **Servers**.
3. Click the name of the managed server where you want to add the scheduler.disabled=false property.
4. Select **Lock and Edit**.
5. Select **Configuration**, **Server Start**.
6. In the Arguments box, add the scheduler.disabled=false property, and click **Save**.
7. Click **Activate Change**.

Restart the managed server using node manager so that the newly added property is picked up. Restarting from the Command-Line Interface does not work.

18.2.1.2 Restarting Oracle Identity Manager Managed Servers from the Node Manager

To restart Oracle Identity Manager Managed Servers from the Node Manager:

1. Start the Administration server. To do so:
   a. From your current working directory, go to the *MW_HOME*/user_projects/domains/base_domain/ directory.
   b. Run the following command:

      For UNIX:

      ```
      startWebLogic.sh
      ```

      For Windows:

      ```
      startWebLogic.cmd
      ```

2. Start the Node Manager. To do so:
   a. From your current working directory, go to the *MW_HOME*/wlserver_10.3/server/bin/ directory.
   b. Run the following command:

      For UNIX:

      ```
      startNodeManager.sh
      ```

      For Windows:

      ```
      startNodeManager.cmd
      ```

3. Log in to the WebLogic Administrative Console.
4. On the left panel, select **Environment**, **Servers**.
5. Select Control from the right panel.
6. Select the option where the property is added, and click **Start**.

18.2.1.3 Modifying the Server Side Property for Oracle Identity Manager

To modify the scheduler.disabled system property:

1. Log in to the WebLogic Administrative Console by using the WebLogic administrator credentials.
2. Under Domain Structure, select **Environment**, **Servers**. The Summary of Servers page is displayed.
3. Click the Oracle Identity Manager server name, for example, oim_server1. The settings for oim_server1 is displayed.
4. Click **Configuration**, **Server Start**.
5. In the Arguments box, change the existing property scheduler.disabled = false/true.
6. Click **Save**.
7. Click **Activate Changes**.
8. Restart the Oracle Identity Manager Managed Server.

> **Note:** After modifying the scheduler.disabled system property, you must start the Managed Server by using the Node Manager.

18.3 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
- Creating Custom Scheduled Tasks

18.3.1 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 18–2 lists the predefined scheduled tasks.

Table 18–2 Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
Application Instance Post Delete Processing Job	This scheduled task is used to revoke, delete, or decommision applicaion instances that have been soft-deleted. It can be run in the following modes: ■ **Revoke:** Deletes the provisioned accounts from the target system after the application instances has been deleted ■ **Delete:** Hard-deletes the accounts from all provisioning tasks and targets, and subsequently from Oracle Identity Manager ■ **Decommission:** Changes the account status to Revoke without keeping the accounts in Oracle Identity Manager in provisioned state	None	Yes
Attestation Grace Period Expiry Checker	This scheduled task delegates the attestation process after the grace period expires.	None	Yes
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	Yes

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
Automatically Unlock User	This scheduled task automatically unlocks a user after the specified number of days. This job supports job frequency in days, minutes, and hours. As password policy in supports lockout duration in minutes, It is recommended to keep the frequency of this scheduled job in minutes.	None	Yes
Bulk Load Archival Job	This scheduled task cleans up the processed entries in the Oracle Identity Manager Database staging tables used during bulk load post processing.	■ Archival Date: This attribute specifies the date up to which the records are purged. It must have a value. The format is **ddMMyyyy** or **MMM dd, yyyy**. ■ Batch Size: Database records are cleaned up in batches. This attribute specifies the size of the batch and must have a value. The default is 1000.	No

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
Bulk Load Post Process	This scheduled task starts post processing jobs for the Bulk Load Utility.	■ Batch Size for Processing Records: User records are processed in batches. This attribute specifies the size of the batch and must have a value. The default is 500. ■ Generate Password: This attribute specifies whether a password is automatically generated when users are created with the Bulk Load Utility. It must have a value of Yes or No; the default is Yes. ■ Ldap Sync: This attribute specifies whether users created in Oracle Identity Manager using the Bulk Load Utility will also be created in the LDAP repository in an LDAP enabled environment. This attribute must have a value of Yes or No; the default is No. ■ Notification: This attribute specifies whether users created using the Bulk Load Utility is notified with an email. It must have a value of Yes or No; the default is Yes. ■ Process User Ids: This attribute specifies the range of user keys (in the Oracle Identity Manager Database) that need to be processed. The keys are associated with the users created using the Bulk Load Utility. It defines a range from start (From:) to finish (To:).	No

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
Catalog Synchronization Job	The scheduled task is used to harvest roles, application instances, and entitlements into the catalog. It is also used to load catalog metadata.	**Mode:** The Catalog Synchronization Job scheduled job can be run in the following modes: ■ **Incremental:** Updates catalog entries based on the Update Date parameter. Only data changed on or after this date is refreshed in the catalog. ■ **Full:** Refreshes the entire catalog from the source entities. All the data in the catalog is replaced. ■ **Metadata:** Updates or adds metadata columns of catalog items based on the supplied CSV file. The CSV file should contain details of the existing catalog items. It should contain Catalog_ID or ENTITY_TYPE, ENTITY_KEY of the existing catalog item. ■ **Technical Glossary:** Loads data in the catalog that represent hierarchical attributes of entitlements based on external source (XML). ■ **Recalculate Tags:** Refreshes CATALOG TAGS column using CATALOG.USER_DEFINED_TAGS and other searchable CATALOG attributes. The same values can be used in keyword search.	Yes
Certification Event Trigger Job	This scheduled task is responsible for running event listeners against the set of user modification events that have occurred in the system. All event listeners are executed by default if none are listed in the Event Listener Name List parameter. See "Configuring Event Listeners and Certification Event Trigger Jobs" in *Performing Self Service Tasks with Oracle Identity Manager* for more information.	Event Listener Name List: This is a comma-separated list of event listeners to be evaluated. If no value if specified for this attribute, then all event listeners are evaluated.	No
DataCollection Scheduled Task	This scheduled task is used to populate data from Oracle Identity Manager operational tables to the staging tables in an offline manner. The scheduled task is set to run manually, and is triggered when Oracle Identity Analytics (OIA) invokes the DataCollectionOperationsIntf->startDataCollection API.	None	Yes

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
Delayed Delete User	This scheduled task automatically deletes the user whose delete date is before the start of today. The XL.UserDeleteDelayPeriod system property indicates the number of days after which the user is to be deleted. When the administrator deletes a user, the user is marked in the Disabled state, and the user's 'Automatically Delete On' date is set for the future date after the number of days indicated in the XL.UserDeleteDelayPeriod system property. This scheduled task finds all such users for whom the 'Automatically Delete On' date is less than the start of today. All those users are marked as Deleted. For example, Jane Doe is a user with '2014-03-24 01:55:00' as the 'Automatically Delete On' date, and John Doe is a user with '2014-03-25 18:55:00' as the 'Automatically Delete On' date. When the scheduler is run on '2014-03-25', only Jane Doe is deleted. John Doe is deleted when the scheduler runs on '2014-03-26'. **Note:** See "System Properties in Oracle Identity Manager" on page 20-1 for information about the XL.UserDeleteDelayPeriod system property. **Note:** Oracle recommendation is to run this scheduled task once per day.	None	No
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. **Note:** Oracle recommendation is to run this scheduled task every 30 minutes or 1 hour.	None	Yes
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. These users are enabled thorough this scheduled task, thereby making the users ACTIVE.	None	Yes
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.	No

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
Entitlement List	This scheduled task populates Entitlement schema from the lookup table whose child process form field `Entitlement` is marked as true.	Auto Publish: When the value of this field is true, the entitlement is automatically published to the organization that is already part of the application instance. The default value of this field is true. If the value is false, then the entitlement is not published to the organization that is already part of the application instance.	No
Entitlement Post Delete Processing Job	This scheduled task is used for post-processing of entitlement soft deletion in the provisioning component. It is used to revoke or delete entitlements that have been soft-deleted. It can be run in the following modes: ■ **Revoke:** Revokes the entitlement-grant for all the accounts in Oracle Identity Manager, which have that specific entitlement granted. ■ **Delete:** Hard-deletes the entitlements from the UD_CHILD table. Irrespective of the mode, the entitlement grant entry is removed from the ENT_ASSIGN table.	None	Yes
Evaluate User Policies	This scheduled task 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.	Yes
Form Upgrade Job	This scheduled task updates the form version to the latest active version and the form data to the value specified during the field's creation for all accounts. **Note:** If this scheduled task is not run, then the form version and data is incorrect in the audit snapshot and the reporting tables.	■ Application Instance Name: Name of the application instance. The default value is "ALL." ■ Batch Size: Use this attribute to fetch number of records from the database to be processed in one iteration. The default value is 500.	Yes

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
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	No
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	No
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.	Yes
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.	No
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	No

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
OIM Certification Purge Job	This scheduled task is used to purge data from the certification tables. It provides for some critical parameters to be specified or configured (although default values are available for these), such as retention period, run duration, and purge criteria, for online and continuous purge of data in the background. **Note:** OIM Certification Purge Job is available in Oracle Identity Manager only if you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170418. For instructions on downloading and applying the 11.1.2.3.161018 bundle patch, refer to the bundle patch documentation. **Note:** By default, the OIM Certification Purge Job is seeded with default values for input parameters, such as purge interval and purge retention period. You must revisit the input parameters to change their default values as required.	For information about the user-configurable attributes, see Configuring Real-Time Certification Purge Job.	No You must manually seed the job and set to Enabled or Disabled.
OIM Data Purge Job	This scheduled task is used as a single unified interface for archive/purge of data for the Requests, Reconciliation,Provisioning Tasks, and Orchestration entities. It provides for some critical parameters to be specified/configured (although default values are available for these), such as retention period, run duration, and purge criteria, for online and continuous purge of data in the background. **Note:** By default, the OIM Data Purge Job scheduled job is seeded in the enabled state with a retention period of 90 days. You must revisit the job parameters to disable or to change the purge interval as required.	For information about the user-configurable attributes, see "Configuring Real-Time Purge and Archival" on page 24-5.	Yes
Password Expiration Task	This scheduled task sends e-mail to users whose password expiration date had passed at the time when the task was run and then updates the USR_PWD_EXPIRED flag on the user profile.	Email Definition Name: Name of the email definition created in the Design Console for sending password expired notification to the user. The default value is "Password Expired".	Yes
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: Name of the email definition created in the Design Console for sending password expiration warning notification to the user. The default value is "Password Expiration Warning".	No
Process Pending Role Grants	This scheduled task is responsible for processing of future role grants. It grants the role for which start date has reached and revokes the role if role grant end date has reached. This task is scheduled to run daily.	None	Yes

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
Reconciliation Retry Scheduled Task	This scheduled task processes the failed reconciliation event for the users whose status is set as Failed.	None	Yes
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	No
Refresh Organization Memberships	This evaluates the organization memberships and assigns users to organizations based on rules. This job evaluates all the organizations whose membership rules have changed since the last job run and their immediate evaluation have not been opted by the administrator.	None	Yes
Refresh Role Memberships	This evaluates the role memberships and assigns users to roles based on rules. This job evaluates all the roles whose membership rules have changed since the last job run and their immediate evaluation have not been opted by the administrator.	None	Yes
Remove Audit Log Entries	This scheduled task is used to permanently remove audit log events which are older than a specified number of days. On job completion, the scheduled task will add a single audit log event in AUDIT_EVENT table recording the number of records removed from the database, the job return code, and an error message if the job fails. For more information on how to control audit data growth in Lightweight audit framework, see "Audit Data Growth Control Measures in Lightweight Audit Framework" on page 24-24.	▪ Batch Size: The number of records to be removed as a batch. Default value is 500. ▪ Maximum Job Duration (in Mins): Default value is 30 minutes. ▪ Remove Audit Log Events older Than (in days): Audit events whose date is older than this value is permanently deleted from the audit event table. Default value is 180 days.	Yes
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 System Administration.	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.	No
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 hours.	Yes

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
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	No
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	No
Retry Failed Orchestrations	This scheduled task retries all failed orchestrations based on the attribute values provided. If there is no parameter value defined, no orchestration is retried.	■ Orchestration ID: This attribute takes a comma separated list of Orchestration Ids to be retried. ■ Entity Type: Orchestrations submitted for the given Entity is retried. ■ Operation: Orchestrations submitted for given Operation is retried. ■ Stage: Orchestrations on the given stage is retried. ■ From Date: Orchestrations submitted after the given date is retried. The format is **ddMMyyyy** or **MMM dd, yyyy**. ■ To Date: Orchestrations submitted before given date is retried. The format is **ddMMyyyy** or **MMM dd, yyyy**.	No
Retry Reconciliation Batch Job	This scheduled task is used to re-process batches with the 'Ready for Processing' status.	Batch ID: This is the comma-separated ID of the batches to be retried.	No
Risk Aggregation Job	This scheduled task is used for calculating the risk summary value for users, roles, and accounts based on their item-risk and risk-factor levels as defined in the system **Note:** See "Understanding Risk Aggregation and Risk Summaries" in *Performing Self Service Tasks with Oracle Identity Manager* for more information.	■ Number of Concurrent Threads: Use this attribute to specify the number of threads that process risk aggregation. ■ User Batch Size: Use this attribute to specify the number of users that must be processed in each thread.	No
Run Future Dated Reconciliation Events	This scheduled task processes the current dated reconciliation event for the users whose status is set as Deferred.	None	No

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
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 scheduled task was run, the task sets the deprovisioned date as the current date.	None	Yes
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	Yes
Seed Home Organization	This scheduled task evaluates and updates organization data for existing users based on configured Home Organization Policy. For more information, see "Managing Home Organization Policy" on page 14-1. Ensure that Home Organization Policy rule for organization evaluation is configured correctly, and the organization should already exist in Oracle Identity Manager. This job can be run for environments that are based on LDAP synchronization. For information about LDAP synchronization, see "Enabling LDAP Synchronization in Oracle Identity Manager" in *Oracle Fusion Middleware Integration Guide for Oracle Identity Management Suite*. Example scenario for LDAP synchronization: During first time identity data sync from the directory server to Oracle Identity Manager, you want to sync organizations based on a rule, which is based on, say department number. To do so: 1. Run the User Create/Update Full Reconciliation scheduled job. This creates users with default organizations provided within the job parameter. 2. Create a home organization rule, and run the Seed Home Organization scheduled job with `Reset Home Organization` option as **Yes**. This overwrites organizations based on the configured rule. **Note:** Run the Seed Home Organization scheduled job with `Reset Home Organization` option as **Yes** with caution because organizations is overwritten.	Batch Size: Use this attribute to fetch number of entries from the persistent store in each query. Reset Home Organization: Use this attribute to determine if the organization value of default users are re-evaluated and overwritten. Select one of the following options: ▪ **No:** If the requirement is to set the organization value for users that do not have any value. ▪ **Yes:** If the requirement is to reset the organization value for all users. This re-evaluates and overrides the organization value for all nondefault users. This option re-evaluates the rule for all existing user data and resets the organization value. If you run the scheduled job with this option selected, then data is overwritten. The **No** option is the default for this scheduled job.	No

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
Sunrise of Accounts and entitlements	This scheduled task sets the status of an account to ENABLE when the start date of the account is reached. In the case of entitlements, this scheduled task grants an entitlement to an account when the start date of the entitlement is reached. **Note:** This task impacts only the accounts and entitlements provisioned directly or through a request.	■ Application Instance Name: Name of the application instance. The default value is "ALL." ■ Max Execution Time: Use this attribute to specify time in minutes, after which the schedule task will stop. The default value is empty. ■ Process Entity Types: Use this attribute to specify whether the task should process accounts or entitlements. The default value is "ALL."	Yes
Sunset of Accounts and entitlements	This scheduled task sets the status of an account to REVOKE or DISABLE when the end date of the account is reached. In the case of entitlements, this scheduled task revokes an entitlement from an account when the end date of the entitlement is reached. **Note:** This task impacts only the accounts and entitlements provisioned directly or through a request.	■ Account Sunset Action: Use this attribute to specify whether the status of the accounts should be set to REVOKE or DISABLE. The default value is REVOKE. ■ Application Instance Name: Name of the application instance. The default value is "ALL." ■ Max Execution Time: Use this attribute to specify time in minutes, after which the schedule task will stop. The default value is empty. ■ Process Entity Types: Use this attribute to specify whether the task should process accounts or entitlements. The default value is "ALL."	Yes
Task Escalation	This scheduled task escalates pending tasks whose escalation time had elapsed at the time when the scheduled task was run.	None	Yes

Table 18–2 (Cont.) Predefined Scheduled Tasks

Job Name	Description	User-Configurable Attributes	Enabled By Default
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	Yes
Update Accounts with App Instance Job	This scheduled task is used to ensure that application instance keys are populated for all entries in the OIU table. In some instances, the application instance might not be available when the account is provisioned. This is possible when: ▪ Oracle Identity Manager is upgraded, when app_instance_key is to be populated for all the existing entries in the OIU table. ▪ Accounts are brought in via reconciliation, but the application instances are not available when the accounts are reconciled. The application instances are created after the reconciliation. ▪ Accounts are provisioned via access policies, but the application instances are not available when the accounts are provisioned. The application instances are created after the provisioning. The Update Accounts with App Instance Job scheduled task checks all the entries in the OIU table corresponding to the resource objects that have a null app_instance_key. It attempts to determine the application instance key based on the obj_key and the IT Resource instance value in the process form. If the scheduled task finds an application instance corresponding to the obj_key and IT resource instance value, then it updates the app_instance_key in the OIU table.	None	Yes
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.	No

18.3.2 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 18–2.

> **See Also:** "Developing Scheduled Tasks" in *Developing and Customizing Applications 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. Place the ZIP file in the file store (the *OIM_HOME*/plugins/ directory) or database store.

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 Developing and Customizing Applications for Oracle Identity Manager* for information about running the UploadJar utility

18.4 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

18.4.1 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 System Administration with the appropriate credentials.
2. In the left pane, under System Configuration, click **Scheduler**. The Advanced Administration is displayed with the Scheduler section in the System Management tab active.
3. On the left pane, from the **Actions** menu, 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:** Retry count is used to manage the job in case of failure. A job cannot execute more than its retry count if it fails consecutively. The job is disabled if it fails consecutively till its retry count is exhausted. The job must be enabled from the UI for further execution.
   - **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.

18.4.2 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

18.4.2.1 Performing a Simple Search for Jobs

To perform a simple search for jobs:

1. In the Welcome page of the Advanced 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.

18.4.2.2 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 18–3 lists the columns of the search results table:

Table 18–3 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

18.4.3 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 18-21 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 18-24 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:

  1. From the View list, select **Reorder Columns.**
  2. In the Reorder Columns dialog box, select the column name that you want to move.
  3. 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:

  1. From the View list, select **Columns.**
  2. Depending on your requirement, select one of the following:

     - Show All

- Start Time

- End Time

- Job Status

- Execution Status

3. Repeat Steps 1 and 2 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 18-24 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**.

18.4.4 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 18-22 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 18-20 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.

18.4.5 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 18-21.
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.

18.4.6 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 18-21.
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.**

18.4.7 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 18-21.

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 cross 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 **Yes.** A message indicating that the job has been deleted successfully is displayed.

18.5 Diagnosing Scheduled Jobs

This section describes how to diagnose issues related to scheduled job run.

Problem

Scheduled job is not running according to the scheduled time, and the following is observed:

- Scheduled job is not run on the scheduled time.
- No entry exists in JOB_HISTORY table for this run. This can be verified by opening the job details in the Scheduler section of Identity System Administration.
- No exceptions are recorded in the server logs.

Solution

To diagnose this issue:

1. Verify whether scheduler service is running or not. Scheduler service is deployed on each node of the cluster until this service is not explicitly disabled. This can be disabled by setting the `scheduler.disabled` server level property to `false` for that node. The following URL can be used to verify the status of the scheduler service:

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

   In this URL, *OIM_HOST* is the name of the computer hosting the Oracle Identity Manager server and *OIM_PORT* is the port on which Oracle Identity Manager server is listening.
2. Verify whether the specific job is enabled or not. This can be verified from the Scheduler section of Identity System Administration. The job must be enabled to run per the schedule.
3. Verify whether clocks are in sync for all nodes. Clocks must be within a second of each other
4. Delete the existing trigger from Scheduler UI, and schedule a new trigger from the UI. Verify whether the issue persists or not.
5. Enable scheduler logs by changing log level to `DEBUG`. This can be done by changing log level for the `oracle.iam.scheduler.impl` package from Oracle Enterprise Manager. Verify whether the following messages are traced in logs or not:

   ```
   Job Listener, Job was executed '$JOB_NAME'
   Job Listener, Job to be executed '$JOB_NAME'
   ```

   Here, $*JOB_NAME* is the name of the job that is supposed to be executed at that time.

   If the messages are not logged, then contact Oracle Support.

6. In Oracle Enterprise Manager, check the `threadPoolSize` parameter for the `schedulerConfig` segment in the `oim-config.xml` file. This is the number of threads that are available for concurrent execution of jobs. Therefore, the number of jobs that can be executed on a particular time cannot be more than the configured `threadPoolSize` count. Running of such jobs is skipped and executed as per the next scheduled time, which gives an impression that the job is not executed per the scheduled time. The default value of this parameter is `10`, but is can be tuned as required.
7. Restart the server and verify whether the job has been run or not.
8. Verify whether the following exception is logged:

   ```
   Caused By: java.lang.NullPointerException at
   org.quartz.SimpleTrigger.computeNumTimesFiredBetween(SimpleTrigger.java:800)
   ```

   Run following query to fix this issue:

   ```
   UPDATE QRTZ92_TRIGGERS SET NEXT_FIRE_TIME=1 WHERE  NEXT_FIRE_TIME<1;
   ```

9. Sometimes the trigger status is not updated in the `QRTZ92_TRIGGER` table from `BLOCKED` to `PAUSED` state. This situation happens if the environment is not tuned properly, and database connections from the pool are exhausted by other parallel operations running on the server. As a result, QUARTZ framework is not able to get connection from the pool to update the running job. This situation can be identified if exceptions related to database connection pool is observed in the server logs. Usually, such triggers get fixed after server restart, but if trigger status still remains the same, then running the following query can help:

   ```
   UPDATE QRTZ92_TRIGGERS SET TRIGGER_STATUS='WAITING' WHERE JOB_NAME
   ='$JOB_NAME'
   ```

   Replace $*JOB_NAME* with the job name.

10. Sometimes manual trigger for a job is not updated in the `QRTZ92_TRIGGER` table. Manual trigger is created in the system when you execute the job by clicking **Run Now** from the Scheduler UI or use the Scheduler `runNow()` API. Such trigger is supposed to be deleted after the job is executed successfully. To fix this issue:
    a. Shutdown the server.
    b. Run the following queries on Oracle Identity Manager database:

       ```
       DELETE FROM QRTZ92_FIRED_TRIGGERS where TRIGGER_NAME like ('MT_%');
       DELETE FROM QRTZ92_SIMPLE_TRIGGERS where TRIGGER_NAME like ('MT_%');
       DELETE FROM QRTZ92_TRIGGERS where TRIGGER_NAME like ('MT_%');
       ```

    Automatic deletion of such manual triggers is maintained by the Quartz framework.

19

Managing Notification Service

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 via generation of errors. Event definition is the metadata that describes the event. To define metadata for events, it is important to identify all event types supported by a functional component. For example, as a part of the scheduler component, metadata can be defined for scheduled job execution failed and shutting down of the scheduler. Every time a job fails or the scheduler is shut down, the events are raised and notifications associated with that event are sent.

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 decide which event variables can be made available at 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. Examples of such channels are e-mail, Instant Messaging (IM), Short Message Service (SMS), and voice. To use these notification providers, Oracle Identity Manager uses Oracle User Messaging Service (UMS).

At the backend, the notification engine is responsible for generating the notification, and utilizing the notification provider to send the notification.

The notification templates and notification providers are described in the following sections:

19.1 Managing Notification Providers

Managing notification providers is described in the following sections:

- Using UMS for Notification
- Using SMTP for Notification
- Using SOA Composite for Notification
- Configuring Custom Notification Provider
- Disabling and Enabling Notification Providers

19.1.1 Using UMS for Notification

UMS offers various capabilities for sending notifications. These capabilities are used by Oracle Identity Manager notification engine to achieve the following:

- **Support for a variety of messaging channels:** Messages can be sent and received through e-mail, IM, SMS, and voice. Oracle Identity Manager supports sending notification messages only via e-mail.
- **Robust message delivery:** UMS keeps track of delivery status information provided by messaging gateways, and makes this information available to applications so that they can respond to a failed delivery.

This section contains the following topics:

- Enabling Oracle Identity Manager to Use UMS for Notification
- Applying OWSM Policy to the UMS Web Service
- Changing UMS Client Connection Pooling

19.1.1.1 Enabling Oracle Identity Manager to Use UMS for Notification

To enable Oracle Identity Manager to use UMS for notification:

1. Configure UMS properties by using the UMSEmailNotificationProviderMBean MBean. To do so:
   a. Log in to Oracle Enterprise Manager.
   b. Click **Application Deployments**.
   c. Right-click **OIMAppMetadata(11.1.2.0.0)(*oim_server_name*)**, and select **System MBean Browser**.
   d. In the System MBean Browser, navigate to **Application Defined MBeans**, **oracle.iam**, **Server:** ***oim_server_name***, **Application: oim**, **IAMAppRuntimeMBean**, and select **UMSEmailNotificationProviderMBean**.
   e. In the Attributes tab, enter the following information:
      * **Policies:** The Messaging UMS web service is used for integration between Oracle Identity Manager and UMS. This Web Service can be secured via Oracle Web Services Manager (OWSM) policy. If OWSM policy is attached to the Messaging web service at server level, then provide the name of the corresponding client side policy. Otherwise, leave the field blank. For example, if oracle/wss11_username_token_with_message_protection_service_policy is applied at the server level, then provide the corresponding client policy name here, such as oracle/wss11_username_token_with_message_protection_client_policy.

* **WSUrl:** This is the URL of the UMS Web service to be started. By default, it contains the URL of the Messaging UMS web service used for integration between Oracle Identity Manager and UMS.

  You can use any other SOA server, for example:

  http://*SOA_HOST*:*SOA_PORT*/ucs/messaging/webservice

  Here, replace *SOA_HOST* with the host name of the SOA server and *SOA_PORT* with the port number to connect to the SOA server.

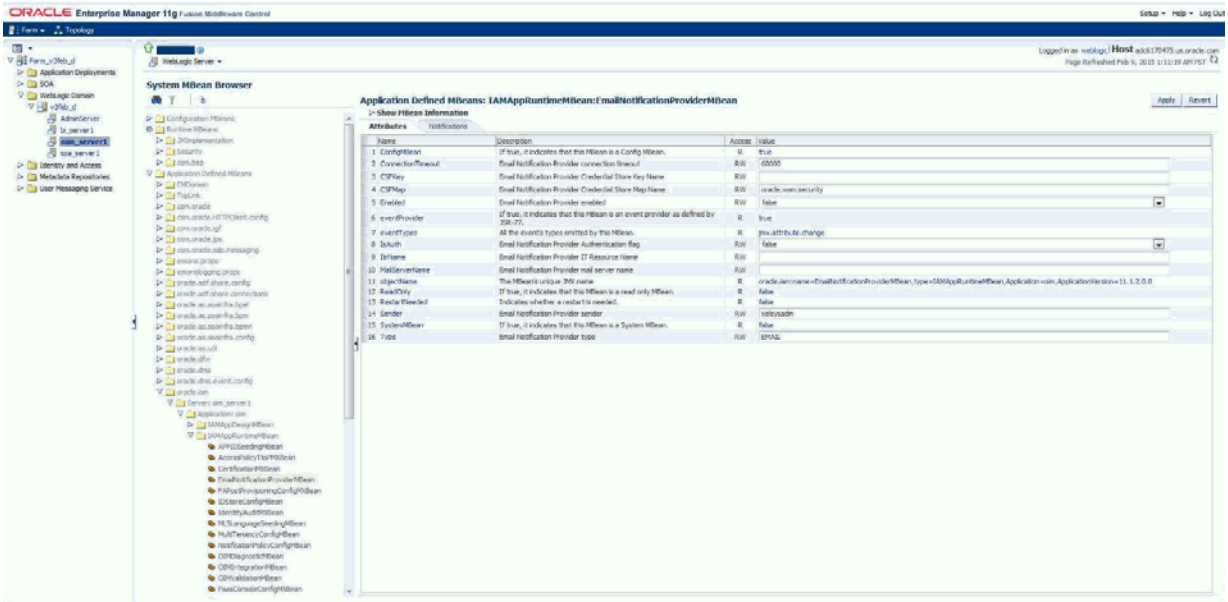
* **CSFKey:** This is the UMS e-mail notification provider credential store (CSF) key name. The key name is populated by default. This key is in the oracle.wsm.security map.

Note: You can find the oracle.wsm.security map as follows:

1. In Oracle Enterprise Manager, expand **WebLogic Domain**.
2. Right-click the base domain, and select **Security**, **Credentials**. The Credentials page is displayed.
3. In the Credential column, expand the oracle.wsm.security map.

Figure 19–1 shows the properties of the UMSEmailNotificationProviderMBean in the Attributes tab of the System MBean Browser.

Figure 19–1 UMSEmailNotificationProviderMBean Properties



f. Click **Apply**.

2. If Oracle Identity Manager and UMS server are in different domains, then you must import the UMS public key into Oracle Identity Manager domain's keystore, and must import Oracle Identity Manager domain's public key into the UMS keystore.

See Also: "Configuring Oracle User Messaging Service" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite* for details about UMS Web service security

3. Configure the mail server. UMS uses the local LINUX mail server by default, and no configuration change is required in UMS for configuring this mail server. However, to use any other Simple Mail Transfer Protocol (SMTP) server:
   a. In Oracle Enterprise Manager, expand **User Messaging Service**, and select **usermessagingdriver-email (*soa_server_name*)**.
   b. From the User Messaging Email Driver list, select **Email Driver Properties**.
   c. In the Driver-Specific Configuration section, populate the following mandatory fields:
      * **OutgoingMailServer:** The name of the SMTP server, for example, stbeehive.oracle.com.
      * **OutgoingMailServerPort:** The port number of the SMTP server, for example, 465.
      * **OutgoingMailServerSecurity:** The security setting used by the SMTP server. Possible values can be None, TLS, or SSL.
      * **OutgoingUsername:** Any valid username similar to your mail client configuration, such as in the firstname.lastname@xyz.com.
      * **OutgoingPassword:** The password used for SMTP authentication. This consists of the following fields:

        Type of Password: Select **Indirect Password, Create New User**.

        Indirect Username/Key: Enter a unique string, for example, OIMEmail-Config. This masks the password and does not expose it in clear text in the configuration file.

        Password: Enter a valid password for this account.
   d. Click **Apply**.
4. If mail server security is SSL, then you must remove DemoTrust store references from the SOA environment. To do so:
   a. In a text editor, open the DOMAIN_HOME/bin/setDomainEnv.sh file. Open setDomainEnv.bat file for Microsoft Windows.
   b. Remove the following line:

      ```
      -Djavax.net.ssl.trustStore=$WL_HOME/server/lib/DemoTrust.jks from
      EXTRA_JAVA_PROPERTIES
      ```

   c. Save and close the file.
   d. In a text editor, open the *DOMAIN_HOME*/bin/startManagedWeblogic.sh file. For Microsoft Windows, open the startManagedWeblogic.bat file.
   e. Remove the following weblogic.security.SSL.trustedCAKeyStore property set in JAVA_OPTIONS from this file:

      ```
      JAVA_OPTIONS="-Dweblogic.security.SSL.trustedCAKeyStore="{MW_HOME}/server/
      server/lib/cacerts" ${JAVA_OPTIONS}"
      ```

   f. Save and close the file.

g. Restart the Admin and Managed servers.

Note: For more details on configuring UMS to connect to a mail server with SSL, see "Configuring Oracle User Messaging Service" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

---

5. Edit the username and password in the CSF key. To do so:
   a. In Oracle Enterprise Manager, expand **WebLogic Domain**.
   b. Right-click the base domain, and select **Security**, **Credentials**. The Credentials page is displayed.
   c. In the Credential column, expand the **oracle.wsm.security** map.
   d. Select the record for the **Notification.Provider.Key** credential key.
   e. On the toolbar, click **Edit**. The Edit Key dialog box is displayed.
   f. Edit the values in the User Name and Password fields, and click **OK**.

19.1.1.2 Applying OWSM Policy to the UMS Web Service

Server-side OWSM policy can be applied to the UMS Web service to protect any other Web service that uses EM. The corresponding client side policy, username, and password must be provided in the provider XML or via MBean.

To attach server-side policy to the UMS Web Service:

1. In Oracle Enterprise Manager, expand **User Messaging Service**, and click **usermessagingserver (soa_server)**.
2. From the User Messaging Service list, select **Web Services**.
3. In the Web Service Details section, click the **Web Service Endpoints** tab.
4. In the Endpoint Name column, click **Messaging**.
5. Click the **OWSM Policies** tab.
6. Under Directly Attached Policies, click **Attach/Detach**. A list of available policies and the options to attach and detach policies are displayed.
7. Select a policy from the available policies list, and click **Attach**. The selected policy is added to the Directly Attached Policies list.

   The policy you select is for securing the Messaging UMS web service.
8. To remove a policy, under Directly Attached Policies, select a policy and click **Detach**. The selected policy is removed from the Directly Attached Policies list.
9. To validate the applied policy combination, click **Validate**. A message is displayed stating that the validation is successful.
10. Click **OK**.

To provide the corresponding client-side policy to the in the provider XML, edit the following properties in the UMS XML Bean in Oracle Identity Manager:

To provide the corresponding client-side policy to the UMSEmailProviderMBean, provide the name of the client-side policy in the UMSEmailNotificationProviderMBean MBean. To do so:

1. Login to Oracle Enterprise Manager.

2. Go to **Application Deployments**. Right-click **OIMAppMetadata(11.1.2.0.0)(oim_server1)**, and select **System MBean Browser**.
3. Go to **Application Defined MBeans**, **oracle.iam, Server: oim_server1**, **Application: oim, IAMAppRuntimeMBean**, **UMSEmailNotificationProviderMBean**.

   Table 19–1 lists the properties of the UMSEmailProviderMBean.

Table 19–1 UMSEmailNotificationProviderMBean Properties

Property	Description
Enabled	A notification provider is used to send the notification e-mail if value for this property is true.
Type	In this release of Oracle Identity Manager, this value is EMAIL only, and the property is not used.
ItrName	Various configuration values required to send the e-mail via UMS, can be either provided in XML properties or IT resource. If configuration values are to be read from IT resource, then provide the name of the IT resource here. If the IT resource name is present, than the IT resource configuration settings are used. If IT resource name is incorrect or invalid, or the values given in the IT resource instance are invalid, then an error is generated and email is not sent. **Note:** Using the IT resource is not a recommended channel to configure UMS in Oracle Identity Manager. This is because there is no mechanism to validate the values provided in the XML or IT resource before sending the e-mail to the server.
WSUrl	The URL of UMS Web service to be invoked. Any SOA server can be used, in the following format: http://*SOA_HOST*/*SOA_PORT*/ucs/messaging/webservice
CSFKey	This is the default notification key under oracle.wsm.security map. This key contains username and password required for OWSM policy. The default and recommended username/password in this key is the WebLogic administrator username and password. This can be changed to any valid username/password on the server side, which is SOA. See step 5 in "Enabling Oracle Identity Manager to Use UMS for Notification" on page 19-2 for information about editing the default values in CSF key by using Oracle Enterprise Mananger.
Policies	If OWSM policy is attached to the given Web service at server level, then provide the name of the corresponding client side policy here. Otherwise, leave this field blank. For example, if oracle/wss11_username_token_with_message_protection_service_policy is applied at server level, then provide the corresponding client policy name here, such as oracle/wss11_username_token_with_message_protection_client_policy.
KeystoreAlias	The keystore alias for the target service. For details about the keystore alias, see "Client Aliases" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.
Sender	A valid username of any Oracle Identity Manager User. The e-mail ID of this user is used to send the e-mail.

4. Provide the client-side policy name in the policies properties shown in this MBean.

19.1.1.3 Changing UMS Client Connection Pooling

For performance enhancement, Oracle Identity Manager creates a pool for UMS client object. The default pool configurations are as follows:

Configuration parameter	Value
abandonedConnectionTimeout	600ms
connectionWaitTimeout	60ms
inactiveConnectionTimeout	300000ms
initialPoolSize	5
maxPoolSize	40
minPoolSize	10
timeoutCheckInterval	30ms
connectionPoolingSupported	true

You can change these default values by passing the following system properties along with new values while starting the Oracle Identity Manager server.

System Properties	Description
ums.ucp.abandonedConnectionTimeout	For abandoned connection timeout
ums.ucp.connectionWaitTimeout	For connection wait timeout
ums.ucp.inactiveConnectionTimeout	For inactive connection timeout
ums.ucp.initialPoolSize	For initial pool size
ums.ucp.maxPoolSize	For maximum pool size
ums.ucp.minPoolSize	For minimum pool size
ums.ucp.timeoutCheckInterval	For timeout check interval
ums.ucp.connectionPoolingSupported	For connection pooling supported

For example, to change initial pool size property (initialPoolSize), you can configure the system property in the following manner:

```
sh startManagedWebLogic.sh oim_server1
 -Dums.ucp.initialPoolSize=10
```

19.1.2 Using SMTP for Notification

By default, the SMTP Email Notification Provider is disabled. This is enabled by setting the value of the enabled attribute to true. To configure SMTP Email Notification Provider properties by using the EmailNotificationProviderMBean MBean:

1. Login to Oracle Enterprise Manager.
2. Click **Application Deployments**.
3. Right-click **OIMAppMetadata(11.1.2.0.0)(oim_server1)**, and select **System MBean Browser**. The System MBean Browser is displayed.

4. Navigate to **Application Defined MBeans**, **oracle.iam**, **Server: oim_server1**, **Application: oim**, **IAMAppRuntimeMBean**, **EmailNotificationProviderMBean**. All the attributes of the EmailNotificationProviderMBean MBean is displayed in the Attributes tab.

Figure 19–2 shows the properties of EmailNotificationProviderMBean in the Attributes tab of the System Mbean Browser.

Figure 19–2 EmailNotificationProviderMBean Properties

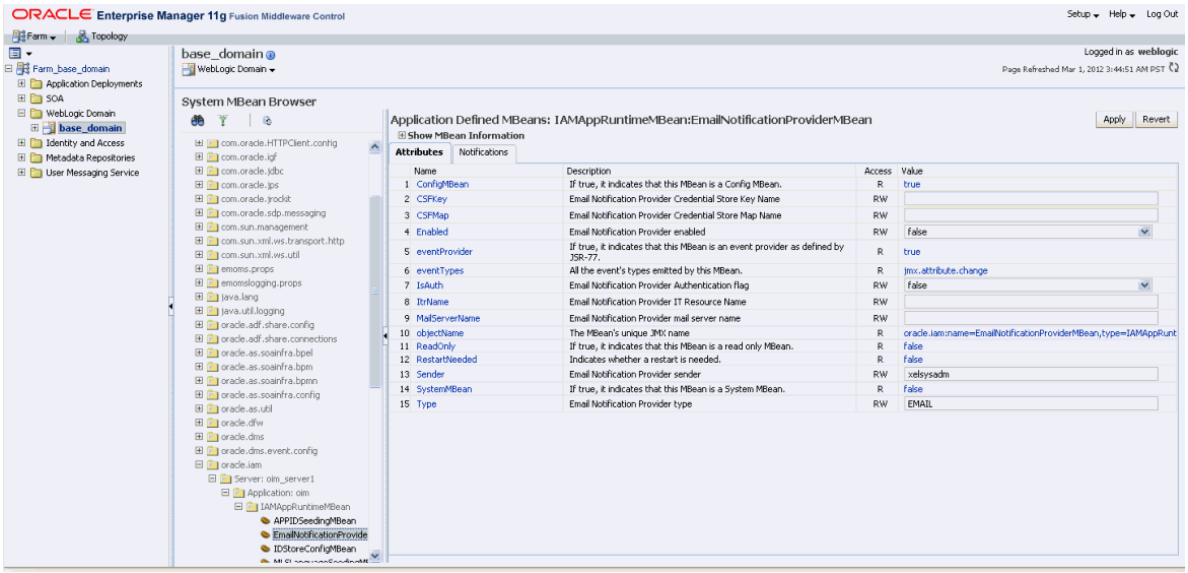


Table 19–2 describes the properties of Default SMTP Email notification provider.

Table 19–2 Default SMTP Email Notification Provider Properties

Property	Description
Enabled	This property derives the status of the notification provider. If the value of this property is false, then the provider is inactive. To activate the provider, change the value to true.
Type	This property determines the type of the notification provider. Oracle Identity Manager supports only Email type.
IsAuth	If the value of this flag is false, then authentication is not required at mail server. As a result, you do not need to provide the CSFKey and CSFMap values. But this depends on the mail server in use. Most of the mail servers support this flag. If any mail server does not support this flag, then authentication credentials must be provided in CSFKey and corresponding CSFMap.
ItrName	If you want to provide connectivity information via IT resource instance of type Mail Server, then provide the name of IT resource instance here. This is not a recommended option.
CSFMap	This property determines the name of the existing CSF Map, for example oim and oracle.wsm.security.

Table 19–2 (Cont.) Default SMTP Email Notification Provider Properties

Property	Description
CSFKey	This property takes the name of the key that contains the authentication credentials, which are username and password. This key must exist under the map name. By default, one key with name Notification.Provider.Key is available under oracle.wsm.security map. This key is used for UMS Email notification provider, and default username and password is weblogic/weblogic1. If UMS email provider is disabled, then use the same map and key to provide the username and password required at mail server for authentication. Otherwise, create a new key under any of the default maps, and provide the name of map and key in these properties. Adding a CSF key is described later in this section.
ConnectionTimeout	This is in milliseconds. This is required for setting a maximum time for connection establishment.
MailServername	This is the name of mail server.
Sender	This is the sender used in Oracle Identity Manager for sending the emails.

To add a CSF key:

1. Login to Oracle Enterprise Manager.
2. Expand **WebLogic Domain**.
3. Right-click **base_domain**, and select **Security**, **Credentials**.
4. Expand **oracle.wsm.security**, and then click **Create Key**.
5. Create a key of type password. Provide the key name, description, username, and password. Click **OK**.

19.1.3 Using SOA Composite for Notification

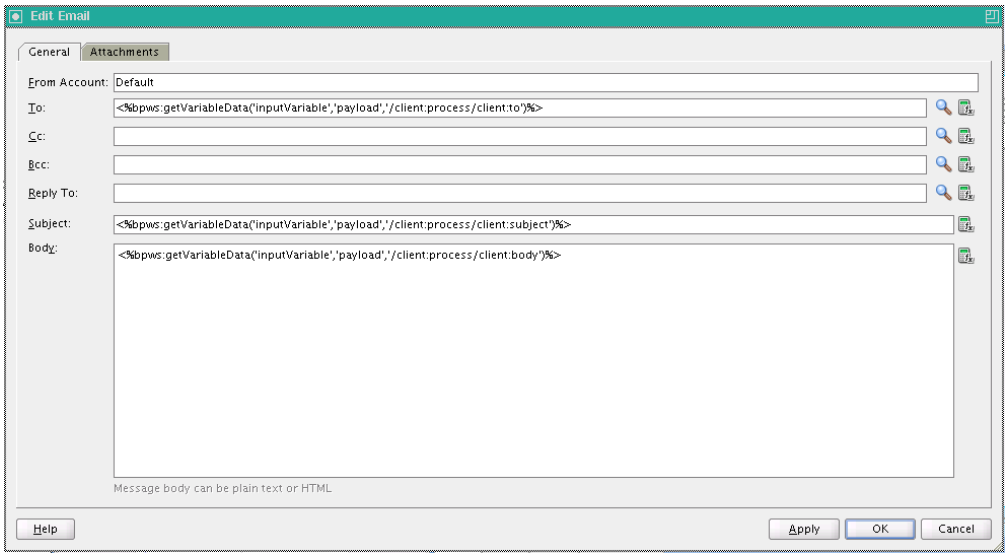
By default, the SOA Email Notification Provider is disabled. You can enable this notification provider by changing the value of the enabled property to true.

To use SOA composite in Oracle Identity Manager for notification:

1. Create a SOA composite with notification activity. For details, see "Using the Notification Service" in the *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

   Figure 19–3 shows the sample mapping of the composite payload via Expression Builder.

Figure 19–3 Sample Mapping of Composite Payload



2. Manually deploy the SOA composite on the SOA server. To do so:

   a. Create an application connection. To do so:

   i. Open the SOA composite in JDeveloper.

   ii. Create a new Application Server Connection by right-clicking the project and selecting **New**, **Connections**, **Application Server Connection**.

   iii. Name the connection as `SOA_server`, and click **Next**.

   iv. Select WebLogic 10.3 as the Connection Type.

   v. Enter the authentication information. The typical values are:

   Username: weblogic

   Password: weblogic1

   vi. On the Connection screen, enter the hostname, port, and SSL port for the SOA Admin server or Admin server, and enter the name of the WebLogic domain.

   vii. Click **Next**.

   viii. On the Test screen, click **Test Connection**. Verify that the success message is displayed.

   b. Deploy the project. To do so:

   i. Right-click the project, select **deploy**, select the project name. Select the **to** option to create the application connection, which is SOA_server. Verify that the build successful message is stored in the log.

   ii. Enter the default revision, and click **OK**. Verify that the Deployment Finished message is stored in the deployment log.

3. Using Enterprise Manager, navigate to **soa-infra**. Right-click **soa-infra**, and select **SOA Administration**, **Workflow Properties**. Under Workflow Notification Properties, select **ALL** from the drop down to set the Notification Mode to ALL.

4. Configure the SOA Email Notification Provider properties by using the SOAEmailNotificationProviderMBean MBean. To do so:

a. Log in to Oracle Enterprise Manager.

b. Expand **Application Deployments**. Right-click **OIMAppMetadata(11.1.2.0.0)(oim_server1)**, and select **System MBean Browser**.

c. Navigate to **Application Defined MBeans**, **oracle.iam**, **Server: oim_server1**, **Application: oim**, **IAMAppRuntimeMBean**, **SOAEmailNotificationProviderMBean**.

5. Change the value of the enabled property from false to true in the SOAEmailNotificationProviderMBean. Figure 19–4 shows the properties of the Bean of SOA Email notification provider.

Figure 19–4 SOAEmailNotificationProviderMBean Properties

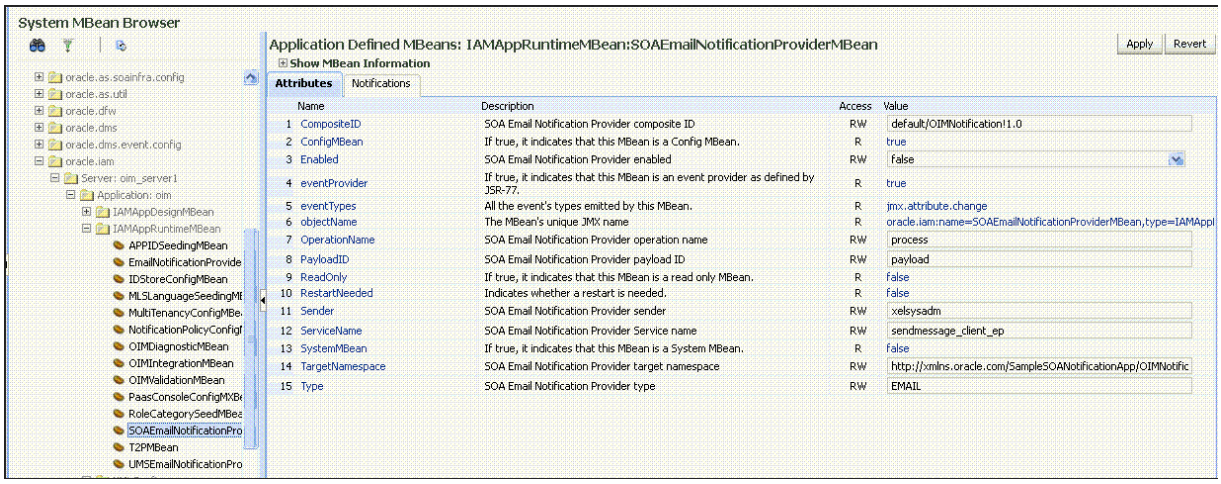


Table 19–3 lists some of the properties of the SOA Email Notification Provider.

Table 19–3 SOA Email Notification Provider Properties

Property	Description
Enabled	This property derives the status of the notification provider. If the value of this property is false, then the provider is inactive. To activate the provider, change the value to true.
Type	This property determines the type of the notification provider. Oracle Identity Manager supports only Email type.
CompositeID	This represents the name of the SOA composite. Name includes pkg/Name!version.
ServiceName	This is the name given to the service in the SOA composite.
OperationName	This is the name given to the process in the SOA composite.
PayloadID	This is the name given to the payload in the SOA composite.
TargetNamespace	This is the name of the targetNamespace given in various XMLs generated while creating the SOA composite.
Sender	This is the sender used in Oracle Identity Manager for sending the emails.

6. Configure the user messaging drivers. if required. If you do not specify values for the user messaging drivers, then the local Linux mail server is used by default. To use any other mail server:

a. Log in to Oracle Enterprise Manager.

b. Navigate to **User Messaging Service**, **usermessagingdriver-email (soa_server1)**, **Email Driver Properties** in Driver-Specific Configuration.

c. Configure the following mandatory values:

- **OutgoingMailServer:** Name of the SMTP server, for example, stbeehive.oracle.com.
- **OutgoingMailServerPort:** Port of the SMTP server, for example, 465.
- **OutgoingMailServerSecurity:** The security setting used by the SMTP server. Possible values can be None, TLS, or SSL.
- **OutgoingUsername:** Any valid username, similar to firstname.lastname@abc.com.
- **OutgoingPassword:** Select **Indirect Password**, **Create New User**. Provide a unique string for Indirect Username/Key, for example, OIMEmailConfig. This will mask the password and not expose it in clear text in the config file. Provide a valid password for this account.

See Also: "Configuring the Email Driver" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management* for more information about configuring user messaging drivers

19.1.4 Configuring Custom Notification Provider

You can configure and use a custom notification provider, other than the default notification providers, for sending notifications.

To configure a custom notification provider:

1. Implement a custom Notification Provider class extending the oracle.iam.notification.provider.NotificationProviderBase base class.
2. Create a JAR file, for example Notification_provider.jar, containing this class.
3. Create an XML file similar to the following:

```
<beans xmlns="http://www.springframework.org/schema/beans"\\\\
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"\\\\
xmlns:util="http://www.springframework.org/schema/util"\\\\
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.spri ngframework.org/schema/beans/spring-beans-2.0.xsd
http://www.springframework.org /schema/util
http://www.springframework.org/schema/util/spring-util-2.0.xsd" def
ault-lazy-init="true">
<bean id="<Name of custom Provider>" class="<Class having custom provider
logic e.g.oracle.iam.notification.provider.CustomProvider>" lazy-init="true">
<!--Mandatory Attributes-->
<property name="enabled" value="<true>" />
<property name="type" value="EMAIL" />
<!--Optional Atributes-->
<property name="sender" value="SYSTEM_ADMINISTRATOR_USERNAME" />
</bean>
</beans>
```

When the value of the enabled property name is true, then this custom provider is used for sending notifications. You can add more properties to this spring bean XML of the custom notification provider, as required.

4. Import the XML file to MDS by using Oracle Enterprise Manager. For information about exporting and importing metadata files to and from MDS, see "Migrating User Modifiable Metadata Files" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.
5. Package all the files as a plug-in.zip file. The structure of the custom notification provider plug-in is:
   - The lib/ directory:

     Notification_provider.jar
   - The plugin.xml file

     See Also: "Developing Plug-ins" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about the concepts related to plug-in and how to develop and use a plug-in

19.1.5 Disabling and Enabling Notification Providers

The notification providers, such as UMS notification provider or EmailNotificationProvider, can be disabled or enabled by using the Enterprise Manager console. For example, to disable UMS notification provider:

1. Login to Enterprise Manager.
2. Go to **Application Deployments**.
3. Right-click **OIMAppMetadata(11.1.2.0.0)(oim_server1)**, and select **System MBean Browser**. The System MBean Browser pane is displayed.
4. Go to **Application Defined MBeans**, **oracle.iam**, **Server: oim_server1**, **Application: oim**, **IAMAppRuntimeMBean**.
5. Select **UMSEmailNotificationProviderMBean**.
6. In the Attributes tab, from the Value list corresponding to the Enabled attribute, select **false** to disable UMS notification provider. To enable UMS notification provider, select **true**.
7. Click **Apply**.

19.2 Managing Notification Templates

Oracle Identity Manager provides a set of default notification templates, as shown in Table 19–4.

Table 19–4 Default Notification Templates

Notification Template	Description
Add Proxy Notification	Template to send notification after a proxy has been added for a user
Bulk Request Creation	Template to send notification during a bulk request creation
Create User Self Service Notification	Template to send notification after a new user is created
End Date	Template to send notification to the manager when end date of the reportee expires
Forgotten Username Notification	Template to send notification after user submits the Forgotten Username form

Table 19–4 (Cont.) Default Notification Templates

Notification Template	Description
Generated Password Notification	Template to send notification after a password is generated by Oracle Identity Manager
Password Expired Notification	Template to send notification after password has expired
Password Warning Notification	Template to send notification before password expires
Request Creation	Template to send notification during a request creation
Request Identity Creation	Template to send notification during a Create User request
Request Status Change	Template to send notification during a request status change
Reset Password	Template to send notification after password has been reset
User Deleted	Template to send notification to the manager when the user account of the reportee is deleted as a result of expired end date

Notification templates are described in the following sections:

- Searching for a Notification Template
- Creating a Notification Template
- Modifying a Notification Template
- Disabling a Notification Template
- Enabling a Notification Template
- Adding and Removing Locales from a Notification Template
- Deleting a Notification Template
- Configuring Notification for a Proxy

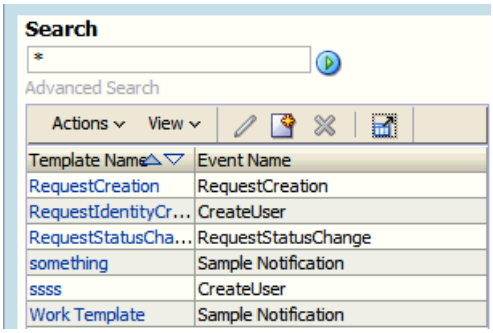
19.2.1 Searching for a Notification Template

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. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Notification**. Advanced Administration is displayed with the Notification tab enabled.
3. Click the icon next to the **Search** field. All the existing notification templates are displayed on the left pane, as shown in Figure 19–5:

Figure 19–5 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 Advanced Administration, click **Advanced Search**. The Advanced Search page is displayed.
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.

19.2.2 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. You must first configure the XML for an event. After this is done, you can create a notification template for that event.

For information about creating the event XML file, see "Defining Event Metadata" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

To create a notification template:

1. Log in to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Notification**. The Notification page is displayed.

3. From the Actions menu, select **Create.** Alternatively, click the Create icon. The Create Template page is displayed.
4. In the Template Information section, enter values for the following fields:
   - **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.

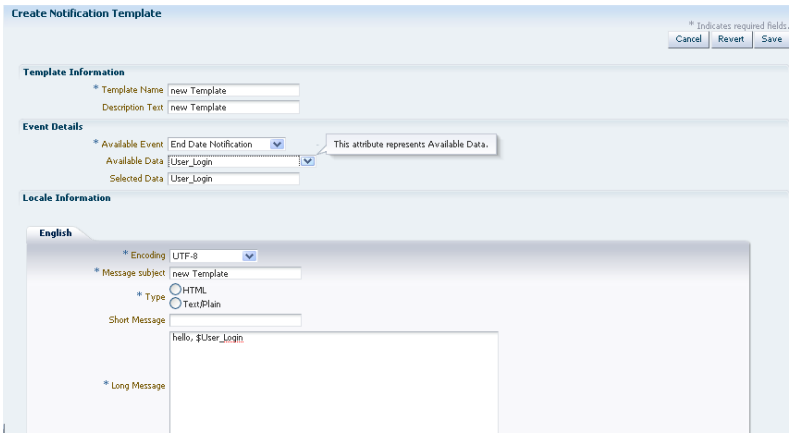
5. In the Event Details section, from the Available Event list, select the event for which the notification template is to be created from a list of available events. Depending on your selection, other fields are displayed in the Event Details section.
6. Under the Locale Information section, enter values 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 short version of the message.
   - In the **Long Message** field, enter the message that is sent as the notification. See step 7.
7. To use the token for available data in the messages that are sent as notification:
   a. In the Event Details section, select the attribute from the Available Data list. This attribute is displayed in the Selected Data field.
   b. Copy the attribute and add it in the message text by placing it inside ${}. For example, if selected data is FA_Territory, then include it in the text as ${FA_Territory}.

   Figure 19–6 shows the Create Notification Template page with sample values:

Figure 19–6 The Create Notification Template Page



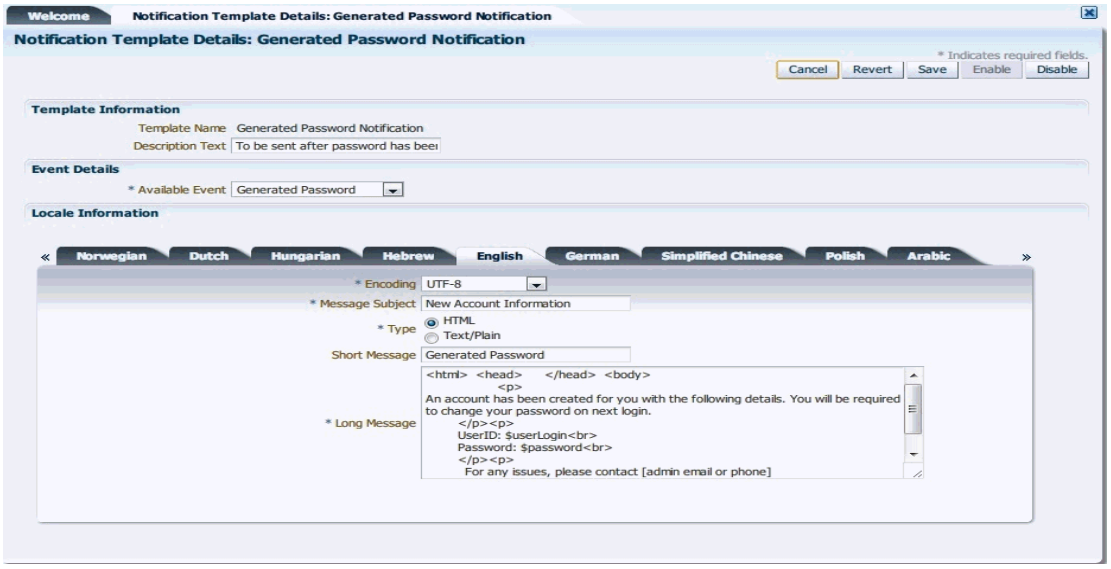
8. After you have entered the required values in all the fields, click **Save.**
9. A message is displayed confirming the creation of the notification template. Click **OK.**

19.2.3 Modifying a Notification Template

To modify a notification template:

1. In Identity System Administration, under System Configuration, click **Notification**.
2. Search for the notification template that you want to modify.
3. Select the template that you want to modify. The the details of a notification template is displayed, as shown in Figure 19–7.

Figure 19–7 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.**

19.2.4 Disabling a Notification Template

You can disable an enabled notification template in the following ways:

- Disable by selecting the notification template in the notification search results. To do so:
  1. In the Identity System Administration, under System Configuration, click **Notification**.
  2. Search for the notification template that you want to disable.
  3. Select the template that you want to disable. Note that the Disable button is active if the notification template is in enabled state.
  4. Click **Disable**. A message is displayed prompting you to confirm the disable operation. Click **Yes**. A message is displayed confirming the disable operation.
- Disable by opening the notification template details. To do so:
  1. In the Identity System Administration, under System Management, click **Notification**.
  2. Search for the notification template that you want to disable.
  3. Click the template name to open the template details. In the notification template details page, the Disable button is active if the notification template is in enabled state, as shown in .Figure 19–7.
  4. Click **Disable**. A message is displayed confirming the disable operation.

19.2.5 Enabling a Notification Template

You can enable a disabled notification template in the following ways:

- Enable by selecting the notification template in the notification search results. To do so:
  1. In the Identity System Administration, under System Configuration, click **Notification**.
  2. Search for the notification template that you want to enable.
  3. Select the template that you want to disable. Note that the Enable button is active if the notification template is in disabled state.
  4. Click **Enable**. A message is displayed prompting you to confirm the enable operation. Click **Yes**. A message is displayed confirming the enable operation.
- Enable by opening the notification template details. To do so:
  1. In the Identity System Administration, under System Management, click **Notification**.
  2. Search for the notification template that you want to enable.
  3. Click the template name to open the template details. In the notification template details page, the Enable button is active if the notification template is in disabled state.
  4. Click **Enable**. A message is displayed confirming the enable operation.

19.2.6 Adding and Removing Locales from a Notification Template

To add locales to a notification template:

1. In the Identity System Administration, under System Configuration, click **Notification**.
2. Search and select the template to which you want to add a locale.
3. From the Actions menu, select **Add Locale.** The Add Locale page is displayed.
4. 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, and click **Confirm.**
5. Click **Next.** The Locale Information page is displayed and the locale that you added is displayed as a tab in the page.
6. In the Locale Information section, specify values for all the fields as mentioned in step 6 of "Creating a Notification Template" on page 19-15, 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. Search for the notification template from which you want to remove a locale. Select the template from the search results table.
2. From the Actions menu, click **Remove Locale**. The Remove Locale page is displayed.
3. Click the icon next to the Locale Name field to select a locale from a list . You can remove a locale from a template only if that template contains multiple locales. You cannot remove a locale if it is the only one associated with the template. Click **Save.**
4. 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.

19.2.7 Deleting a Notification Template

To delete a notification template:

1. In the Identity System Administration, under System Configuration, click **Notification**.
2. Search for the notification template that you want to delete.
3. Select the template that you want to delete.
4. From the Actions menu, click **Delete.** Alternatively, click the cross icon on the toolbar. A message is displayed prompting you to confirm the delete the operation. Click **Yes.** A message is displayed confirming the delete operation.

19.2.8 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 user. (For example, create a user Jane Doe.)
3. Create a second user. (For example, create a user John Doe.)
4. Assign the Jane Doe user as a manager for John Doe.
5. Specify your email ID for John Doe, which enables you to receive notifications in your inbox.
6. Login to Oracle Identity Self Service as Jane Doe.
7. In the Self Service tab, click **My Information**. The My Information page is displayed.
8. Expand **Proxies**. In the Proxies section, add John Doe as a proxy for Jane Doe.

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

"You have been made the proxy for Jane Doe [JANED] from April 9, 2012 12:00:00 AM to April 30, 2010 12:00:00 AM".

19.3 Configuring Email in Provisioning Workflow

You can configure email notifications for using them in provisioning processes by configuring the default email provider.

To configure default email provider:

1. Login to Oracle Identity System Administration, and set the value of the Email Server system property (with keyword XL.MailServer) to point to the IT resource with name Email Server. For information about this system property, see "Configuring Oracle Identity Manager" on page 20-1.
2. Verify that the Email Server IT resource exists. This IT resource must have Mail Server as the IT resource type, and it must have a server name, for example localhost. If this IT resource is not present for mail server, then create the IT resource. For information about creating IT resources, see "Creating IT Resources" on page 8-1.

19.4 Configuring SOA Email Notification

This section contains the following topics:

- Configuring Actionable Email Notification on SOA
- Troubleshooting SOA Email Notification

19.4.1 Configuring Actionable Email Notification on SOA

To configure email notifications on SOA:

1. Before performing the steps to configure email notifications in SOA, ensure the following:

- Make sure that the user to whom task is assigned has a valid email account set in Oracle Identity Manager.
- If you want email notifications to be actionable, such as allowing approving or rejecting requests from the email, then ensure that you have configured human task to send actions in the notification. You can verify this by using SOA Composer. To do so:
  a. Login to SOA Composer by using weblogic user by using the following URL:

     http://*SOA_HOST*:*SOA_PORT*/soa/composer
  b. From the Open menu, select **Open Tasks**.
  c. In the Select a Task to open dialog box, select the human task for which you want to verify the settings, and then click **Open**.
  d. In Notification Settings section, verify that the **Make notification actionable** option is selected.

2. Login to Oracle Enterprise Manager as weblogic user.
3. Go to SOA. Right-click **soa-infra (*soa_server_name*)**, and select **SOA Administration**, **Workflow Properties**.
4. In the Workflow Notification Properties dialog box, select Email from the Notification Mode list.
5. Enter values for the following:
   - **Email : From Address:** Email account from which notification is sent to approvers
   - **Email : Actionable Address:** Email account that will receive approve/reject response sent by approvers via email
   - **Email : Reply To Address:** Optional email address to which the reply is sent, for example, no.reply@yourdomain.com
6. Click **Apply**.
7. Go to User Messaging Service. Right-click **usermessagingdriver-email (*soa_server_name*)**, and select **Email Driver properties**.
8. In the Driver-Specific Configuration section, configure the following minimum attributes for email notifications to work correctly:
   - **MailAccessProtocol:** Select IMAP or POP3
   - **OutgoingMailServer:** Name of the SMTP server, for example, myhost.mycompany.com
   - **OutgoingMailServerPort:** Port of the SMTP server, for example, 465
   - **OutgoingDefaultFromAddress:** Same as OutgoingMailServer
   - **OutgoingPassword:** You can provide the password in clear text stored in driverconfig.xml, or store password in CSF by using indirect option.
   - **IncomingMailServer:** The hostname of the incoming mail server. Required only if receiving emails is supported on the driver instance.
   - **IncomingMailIDs:** The email addresses corresponding to the user names. Each email address is separated by a comma and must reside in the same position in the list as their corresponding user name appears on the usernames list. Required only if receiving emails is supported on the driver instance.

- **IncomingUserPasswords:** You can provide password in clear text stored in driverconfig.xml, or store password in CSF using indirect option.
- **Debug (Optional):** Setting this to true logs all email activity on SOA server console but not SOA log files. Set this to true until you are sure that notifications are working correctly.

  See Also: "Configuring Human Workflow Service Components and Engines" and "Configuring Oracle User Messaging Service" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite* for detailed information about driver-specific configuration and Human workflow service components

9. Click **Apply**, and restart SOA Managed Server.

19.4.2 Troubleshooting SOA Email Notification

Consider the following to troubleshoot issues encountered with SOA email notification:

- Enable the Debug option in the driver-specific configuration if you are facing issues with sending or receiving notifications. If you modify the email driver properties, then restart SOA server.
- Send test notifications. To do so:
  a. Login to Oracle Enterprise Manager.
  b. Go to SOA. Right-click **soa-infra (*soa_server_name*)**, and select **Service Engines**, **Human Workflow**, **Notification Management**, **Send Test Notification**.
- Verify that email server and accounts are working. Try sending/receiving emails using your email client.
- Check the SOA server log. Usually, the issue is with user messaging service configuration. If you have enabled the debug option, then SOA server log provides debugging information.
- Sometimes if email is not being sent to a particular email account (because of incorrect configuration), then SOA server marks it as bad address. You must manually remove such bad address. To do so:
  a. Login to Oracle Enterprise Manager.
  b. Go to SOA. Right-click **soa-infra (*soa_server_name*)**, **Service Engines**, **Human Workflow**, **Notification Management**, **View Bad Address**, **Remove the Bad Address**.

19.5 Disabling Oracle Identity Manager Email Notifications

Notifications are sent in the following scenarios by event handlers when users are created through UI or through SPML:

See Also: "Developing Event Handlers" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for information about event handlers

- A user is created with manual password as a result of SelfServiceNotificationHandler. To disable sending email notification, remove the SelfServiceNotificationHandler section in the /metadata/iam-features-selfservice/event-definition/EventHandlers.xml in MDS. To do so:

  1. Export the /metadata/iam-features-selfservice/event-definition/EventHandlers.xml file from MDS by using Oracle Enterprise Manager. See "Migrating User Modifiable Metadata Files" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

     Note: Save a local copy of the EventHandlers.xml for future reference.

  2. Remove the following from the EventHandlers.xml file:

     ```
     <postprocess-handler
     class="oracle.iam.selfservice.uself.uselfmgmt.impl.handlers.create.SelfSer
     viceNotificationHandler"
     entity-type="User"
     operation="CREATE"
     name="SelfServiceNotificationHandler"
     order="1160"
     stage="postprocess"
     sync="TRUE">
     </postprocess-handler>
     ```

  3. Import the files to MDS by following the instructions in "Migrating User Modifiable Metadata Files" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

  4. Export the files again to verify that the edits have been correctly uploaded to MDS.

- System Administrator creates user with autogenerated password as a result of PasswordNotificationHandler. To disable sending email notification, remove the PasswordNotificationHandler section in the /metadata/iam-features-passwordmgmt/event-definition/EventHandlers.xml file in MDS. To do so:

  1. Export the /metadata/iam-features-passwordmgmt/event-definition/EventHandlers.xml file from MDS by using Oracle Enterprise Manager.

  2. Remove the following from the EventHandlers.xml file:

     ```
     <postprocess-handler
     class="oracle.iam.passwordmgmt.eventhandlers.PasswordNotificationHandler"
     entity-type="User" operation="CREATE" name="PasswordNotificationHandler"
     order="1180" stage="postprocess" sync="TRUE">
     </postprocess-handler>
     ```

  3. Import the files to MDS by following the instructions in "Migrating User Modifiable Metadata Files" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

  4. Export the files again to verify that the edits have been correctly uploaded to MDS.

- System Administrator changes password manually. The notification can be disabled through UI based on the email checkbox selected on the UI.
- System Administrator changes password with autogenerated password (reset password) as a result of ResetPasswordActionHandler. This is not a postprocess event handler that can be disabled.
- To disable notifications related to reconciliation, login to Oracle Identity System Administration, and set the 'Should send notifications in recon or not' system property to FALSE. For information about this system property, see "System Properties in Oracle Identity Manager" on page 20-1.
- To disable all email notifications in Oracle Identity Manager, set the value of the XL.DisableAllNotifications system property to true. By default, the value of this system property is false. If an incorrect value is specified for this system property, then notifications are enabled. See "System Properties in Oracle Identity Manager" on page 20-1 for information about this system property.

19.6 Troubleshooting Notification

This section describes the following issues that you might encounter with UMS configuration and the corresponding solutions:

- Issues Related to Incorrect URL
- Incorrect Outgoing Server EMail Driver Properties
- Error Generated at the SOA Server
- Authentication Failure
- Issues Related to Failed Email Delivery Not Reported Through EM

19.6.1 Issues Related to Incorrect URL

Problem

Oracle Identity Manager log shows the following error:

```
<Jun 13, 2012 12:53:25 AM PDT> <Warning>
<oracle.adfinternal.view.faces.renderkit.rich.SelectItemUtils> <ADF_FACES-30118>
<No help provider found for helpTopicId=create_user.>
java.net.MalformedURLException: For input string: "SOA_PORT"
at java.net.URL.<init>(URL.java:601)
at java.net.URL.<init>(URL.java:464)
at java.net.URL.<init>(URL.java:413)
at java.net.URI.toURL(URI.java:1081)
at oracle.j2ee.ws.common.transport.HttpTransport.transmit(HttpTransport.java:61)
at oracle.j2ee.ws.common.async.MessageSender.call(MessageSender.java:64)
at oracle.j2ee.ws.common.async.Transmitter.transmitSync(Transmitter.java:134)
at oracle.j2ee.ws.common.async.Transmitter.transmit(Transmitter.java:90)
at oracle.j2ee.ws.common.async.RequestorImpl.transmit(RequestorImpl.java:273)
at oracle.j2ee.ws.common.async.RequestorImpl.invoke(RequestorImpl.java:94)
at oracle.j2ee.ws.client.jaxws.DispatchImpl.invoke(DispatchImpl.java:811)
at
oracle.j2ee.ws.client.jaxws.OracleDispatchImpl.synchronousInvocationWithRetry(Orac
leDispatchImpl.java:235)
at
oracle.j2ee.ws.client.jaxws.OracleDispatchImpl.invoke(OracleDispatchImpl.java:106)
at
oracle.j2ee.ws.client.jaxws.WsClientProxyInvocationHandler.invoke(WsClientProxyInv
```

```
ocationHandler.java:254)
at $Proxy422.send(Unknown Source)
at oracle.ucs.messaging.ws.MessagingClient.send(MessagingClient.java:299)
at
oracle.iam.notification.provider.UMSEmailServiceProvider.sendMessage(UMSEmailServi
ceProvider.java:188)
at
oracle.iam.notification.provider.UMSEmailServiceProvider.sendNotification(UMSEmail
ServiceProvider.java:173)
at
oracle.iam.notification.impl.NotificationServiceImpl.sendEmailNotification(Notific
ationServiceImpl.java:601)
at
oracle.iam.notification.impl.NotificationServiceImpl.notify(NotificationServiceImp
l.java:540)
at
oracle.iam.notification.impl.NotificationServiceImpl.notify(NotificationServiceImp
l.java:271)
<Jun 13, 2012 12:53:31 AM PDT> <Error> <oracle.iam.notification.impl> <BEA-000000>
<Provider UMSEmailServiceProvider has encountered exception : null>
<Jun 13, 2012 12:53:31 AM PDT> <Error> <oracle.iam.notification.impl> <BEA-000000>
<Sending notification with Provider UMSEmailServiceProvider has encountered
exception : Error occured while Sending Notification through Provider
UMSEmailServiceProvider : null>
<Jun 13, 2012 12:53:31 AM PDT> <Error> <oracle.iam.notification.impl> <BEA-000000>
<Sending notification with Provider UMSEmailServiceProvider detailed exception :
null>
```

Solution

The cause of this error is malformed URL. To resolve the issue, provide the correct values for *SOA_PORT* and *SOA_HOST* in Enterprise Manager (EM).

Problem

Oracle Identity Manager log shows the following error:

```
<Jun 13, 2012 3:14:14 AM PDT> <Error> <oracle.iam.notification.impl> <BEA-000000>
<Provider UMSEmailServiceProvider has encountered exception :
javax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Bad response: 404 Not
Found from url http://myhost.mycompany.com:8003/ucs/messaging/webservice>
<Jun 13, 2012 3:14:14 AM PDT> <Error> <oracle.iam.notification.impl> <BEA-000000>
<Sending notification with Provider UMSEmailServiceProvider has encountered
exception : Error occured while Sending Notification through Provider
UMSEmailServiceProvider : javax.xml.soap.SOAPException:
javax.xml.soap.SOAPException: Bad response: 404 Not Found from url
http://myhost.mycompany.com:8003/ucs/messaging/webservice>
<Jun 13, 2012 3:14:14 AM PDT> <Error> <oracle.iam.notification.impl> <BEA-000000>
<Sending notification with Provider UMSEmailServiceProvider detailed exception :
javax.xml.soap.SOAPException: javax.xml.soap.SOAPException: Bad response: 404 Not
Found from url http://myhost.mycompany.com:8003/ucs/messaging/webservice>
```

Solution

The cause of this problem in incorrect URL. To resolve the issue, provide the correct URL in EM.

19.6.2 Incorrect Outgoing Server EMail Driver Properties

Problem

The following error is generated:

```
<Jun 13, 2012 3:39:14 AM PDT> <Error> <oracle.sdp.messaging.driver.email>
<SDP-25700> <An unexpected exception was caught.
javax.mail.MessagingException: Unknown SMTP host: abc.mydomain.com;
nested exception is:
java.net.UnknownHostException: abc.mydomain.com
at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1389)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:412)
at javax.mail.Service.connect(Service.java:310)
at javax.mail.Service.connect(Service.java:169)
at javax.mail.Service.connect(Service.java:118)
at
oracle.sdpinternal.messaging.driver.email.EmailDriver.send(EmailDriver.java:780)
at
oracle.sdpinternal.messaging.driver.email.EmailManagedConnection.send(EmailManaged
Connection.java:50)
at
oracle.sdpinternal.messaging.driver.DriverConnectionImpl.send(DriverConnectionImpl
.java:41)
at
oracle.sdpinternal.messaging.dispatcher.DriverDispatcherBean.onMessage(DriverDispa
tcherBean.java:296)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
at
com.bea.core.repackaged.springframework.aop.support.AopUtils.invokeJoinpointUsingR
eflection(AopUtils.java:310)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.i
nvokeJoinpoint(ReflectiveMethodInvocation.java:182)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:149)
at
com.bea.core.repackaged.springframework.aop.interceptor.ExposeInvocationIntercepto
r.invoke(ExposeInvocationInterceptor.java:89)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.doProceed(DelegatingIntroductionInterceptor.java:131)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.invoke(DelegatingIntroductionInterceptor.java:119)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.framework.JdkDynamicAopProxy.invoke(Jd
kDynamicAopProxy.java:204)
at $Proxy349.onMessage(Unknown Source)
```

```
at weblogic.ejb.container.internal.MDListener.execute(MDListener.java:583)
at
weblogic.ejb.container.internal.MDListener.transactionalOnMessage(MDListener.java:
486)
at weblogic.ejb.container.internal.MDListener.onMessage(MDListener.java:388)
at weblogic.jms.client.JMSSession.onMessage(JMSSession.java:4659)
at weblogic.jms.client.JMSSession.execute(JMSSession.java:4345)
at weblogic.jms.client.JMSSession.executeMessage(JMSSession.java:3821)
at weblogic.jms.client.JMSSession.access$000(JMSSession.java:115)
at weblogic.jms.client.JMSSession$UseForRunnable.run(JMSSession.java:5170)
at
weblogic.work.SelfTuningWorkManagerImpl$WorkAdapterImpl.run(SelfTuningWorkManagerI
mpl.java:545)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
Caused By: java.net.UnknownHostException: abc.mydomain.com
at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:195)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:366)
at java.net.Socket.connect(Socket.java:529)
at com.sun.net.ssl.internal.ssl.SSLSocketImpl.connect(SSLSocketImpl.java:564)
at
com.sun.net.ssl.internal.ssl.BaseSSLSocketImpl.connect(BaseSSLSocketImpl.java:141)
at com.sun.mail.util.SocketFetcher.createSocket(SocketFetcher.java:233)
at com.sun.mail.util.SocketFetcher.getSocket(SocketFetcher.java:163)
at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1359)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:412)
at javax.mail.Service.connect(Service.java:310)
at javax.mail.Service.connect(Service.java:169)
at javax.mail.Service.connect(Service.java:118)
at
oracle.sdpinternal.messaging.driver.email.EmailDriver.send(EmailDriver.java:780)
at
oracle.sdpinternal.messaging.driver.email.EmailManagedConnection.send(EmailManaged
Connection.java:50)
at
oracle.sdpinternal.messaging.driver.DriverConnectionImpl.send(DriverConnectionImpl
.java:41)
at
oracle.sdpinternal.messaging.dispatcher.DriverDispatcherBean.onMessage(DriverDispa
tcherBean.java:296)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
at
com.bea.core.repackaged.springframework.aop.support.AopUtils.invokeJoinpointUsingR
eflection(AopUtils.java:310)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.i
nvokeJoinpoint(ReflectiveMethodInvocation.java:182)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:149)
at
com.bea.core.repackaged.springframework.aop.interceptor.ExposeInvocationIntercepto
r.invoke(ExposeInvocationInterceptor.java:89)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
```

```
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.doProceed(DelegatingIntroductionInterceptor.java:131)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.invoke(DelegatingIntroductionInterceptor.java:119)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.framework.JdkDynamicAopProxy.invoke(Jd
kDynamicAopProxy.java:204)
at $Proxy349.onMessage(Unknown Source)
at weblogic.ejb.container.internal.MDListener.execute(MDListener.java:583)
at
weblogic.ejb.container.internal.MDListener.transactionalOnMessage(MDListener.java:
486)
at weblogic.ejb.container.internal.MDListener.onMessage(MDListener.java:388)
at weblogic.jms.client.JMSSession.onMessage(JMSSession.java:4659)
at weblogic.jms.client.JMSSession.execute(JMSSession.java:4345)
at weblogic.jms.client.JMSSession.executeMessage(JMSSession.java:3821)
at weblogic.jms.client.JMSSession.access$000(JMSSession.java:115)
at weblogic.jms.client.JMSSession$UseForRunnable.run(JMSSession.java:5170)
at
weblogic.work.SelfTuningWorkManagerImpl$WorkAdapterImpl.run(SelfTuningWorkManagerI
mpl.java:545)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
>
```

Solution

The cause of this problem is incorrect Outgoing Server EMail Driver properties. To rectify the issue, provide the correct email server address, and ensure that the server is running.

19.6.3 Error Generated at the SOA Server

Problem

The following error is displayed in the SOA server logs:

```
<Jun 13, 2012 3:53:20 AM PDT> <Error> <oracle.sdp.messaging.driver.email>
<SDP-25700> <An unexpected exception was caught.
javax.mail.MessagingException: Could not connect to SMTP host:
stbeehive.mydomain.com, port: 25;
nested exception is:
java.net.ConnectException: Connection refused
at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1391)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:412)
at javax.mail.Service.connect(Service.java:310)
at javax.mail.Service.connect(Service.java:169)
at javax.mail.Service.connect(Service.java:118)
at
oracle.sdpinternal.messaging.driver.email.EmailDriver.send(EmailDriver.java:780)
at
oracle.sdpinternal.messaging.driver.email.EmailManagedConnection.send(EmailManaged
Connection.java:50)
at
```

```
oracle.sdpinternal.messaging.driver.DriverConnectionImpl.send(DriverConnectionImpl
.java:41)
at
oracle.sdpinternal.messaging.dispatcher.DriverDispatcherBean.onMessage(DriverDispa
tcherBean.java:296)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
at
com.bea.core.repackaged.springframework.aop.support.AopUtils.invokeJoinpointUsingR
eflection(AopUtils.java:310)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.i
nvokeJoinpoint(ReflectiveMethodInvocation.java:182)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:149)
at
com.bea.core.repackaged.springframework.aop.interceptor.ExposeInvocationIntercepto
r.invoke(ExposeInvocationInterceptor.java:89)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.doProceed(DelegatingIntroductionInterceptor.java:131)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.invoke(DelegatingIntroductionInterceptor.java:119)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.framework.JdkDynamicAopProxy.invoke(Jd
kDynamicAopProxy.java:204)
at $Proxy345.onMessage(Unknown Source)
at weblogic.ejb.container.internal.MDListener.execute(MDListener.java:583)
at
weblogic.ejb.container.internal.MDListener.transactionalOnMessage(MDListener.java:
486)
at weblogic.ejb.container.internal.MDListener.onMessage(MDListener.java:388)
at weblogic.jms.client.JMSSession.onMessage(JMSSession.java:4659)
at weblogic.jms.client.JMSSession.execute(JMSSession.java:4345)
at weblogic.jms.client.JMSSession.executeMessage(JMSSession.java:3821)
at weblogic.jms.client.JMSSession.access$000(JMSSession.java:115)
at weblogic.jms.client.JMSSession$UseForRunnable.run(JMSSession.java:5170)
at
weblogic.work.SelfTuningWorkManagerImpl$WorkAdapterImpl.run(SelfTuningWorkManagerI
mpl.java:545)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
Caused By: java.net.ConnectException: Connection refused
at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:351)
at java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:213)
at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:200)
at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:366)
```

```
at java.net.Socket.connect(Socket.java:529)
at com.sun.net.ssl.internal.ssl.SSLSocketImpl.connect(SSLSocketImpl.java:564)
at
com.sun.net.ssl.internal.ssl.BaseSSLSocketImpl.connect(BaseSSLSocketImpl.java:141)
at com.sun.mail.util.SocketFetcher.createSocket(SocketFetcher.java:233)
at com.sun.mail.util.SocketFetcher.getSocket(SocketFetcher.java:163)
at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1359)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:412)
at javax.mail.Service.connect(Service.java:310)
at javax.mail.Service.connect(Service.java:169)
at javax.mail.Service.connect(Service.java:118)
at
oracle.sdpinternal.messaging.driver.email.EmailDriver.send(EmailDriver.java:780)
at
oracle.sdpinternal.messaging.driver.email.EmailManagedConnection.send(EmailManaged
Connection.java:50)
at
oracle.sdpinternal.messaging.driver.DriverConnectionImpl.send(DriverConnectionImpl
.java:41)
at
oracle.sdpinternal.messaging.dispatcher.DriverDispatcherBean.onMessage(DriverDispa
tcherBean.java:296)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
at
com.bea.core.repackaged.springframework.aop.support.AopUtils.invokeJoinpointUsingR
eflection(AopUtils.java:310)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.i
nvokeJoinpoint(ReflectiveMethodInvocation.java:182)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:149)
at
com.bea.core.repackaged.springframework.aop.interceptor.ExposeInvocationIntercepto
r.invoke(ExposeInvocationInterceptor.java:89)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.doProceed(DelegatingIntroductionInterceptor.java:131)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.invoke(DelegatingIntroductionInterceptor.java:119)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.framework.JdkDynamicAopProxy.invoke(Jd
kDynamicAopProxy.java:204)
at $Proxy345.onMessage(Unknown Source)
at weblogic.ejb.container.internal.MDListener.execute(MDListener.java:583)
at
weblogic.ejb.container.internal.MDListener.transactionalOnMessage(MDListener.java:
486)
```

```
at weblogic.ejb.container.internal.MDListener.onMessage(MDListener.java:388)
at weblogic.jms.client.JMSSession.onMessage(JMSSession.java:4659)
at weblogic.jms.client.JMSSession.execute(JMSSession.java:4345)
at weblogic.jms.client.JMSSession.executeMessage(JMSSession.java:3821)
at weblogic.jms.client.JMSSession.access$000(JMSSession.java:115)
at weblogic.jms.client.JMSSession$UseForRunnable.run(JMSSession.java:5170)
at
weblogic.work.SelfTuningWorkManagerImpl$WorkAdapterImpl.run(SelfTuningWorkManagerI
mpl.java:545)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
>
```

Solution

This is an error in SOA server. To rectify the issue, ensure that the outgoing server host, outgoing server port, and outgoing server security information are provided correctly.

19.6.4 Authentication Failure

Problem

The following errors are generated:

```
javax.mail.AuthenticationFailedException
```

OR

```
<Jun 13, 2012 4:30:41 AM PDT> <Error> <oracle.sdp.messaging.driver.email>
<SDP-25700> <An unexpected exception was caught.
javax.mail.MessagingException: Exception reading response;
nested exception is:
javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException:
PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target
at com.sun.mail.smtp.SMTPTransport.readServerResponse(SMTPTransport.java:1611)
at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1369)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:412)
at javax.mail.Service.connect(Service.java:310)
at javax.mail.Service.connect(Service.java:169)
at javax.mail.Service.connect(Service.java:118)
at
oracle.sdpinternal.messaging.driver.email.EmailDriver.send(EmailDriver.java:780)
at
oracle.sdpinternal.messaging.driver.email.EmailManagedConnection.send(EmailManaged
Connection.java:50)
at
oracle.sdpinternal.messaging.driver.DriverConnectionImpl.send(DriverConnectionImpl
.java:41)
at
oracle.sdpinternal.messaging.dispatcher.DriverDispatcherBean.onMessage(DriverDispa
tcherBean.java:296)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
```

```
at
com.bea.core.repackaged.springframework.aop.support.AopUtils.invokeJoinpointUsingR
eflection(AopUtils.java:310)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.i
nvokeJoinpoint(ReflectiveMethodInvocation.java:182)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:149)
at
com.bea.core.repackaged.springframework.aop.interceptor.ExposeInvocationIntercepto
r.invoke(ExposeInvocationInterceptor.java:89)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.doProceed(DelegatingIntroductionInterceptor.java:131)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.invoke(DelegatingIntroductionInterceptor.java:119)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.framework.JdkDynamicAopProxy.invoke(Jd
kDynamicAopProxy.java:204)
at $Proxy349.onMessage(Unknown Source)
at weblogic.ejb.container.internal.MDListener.execute(MDListener.java:583)
at
weblogic.ejb.container.internal.MDListener.transactionalOnMessage(MDListener.java:
486)
at weblogic.ejb.container.internal.MDListener.onMessage(MDListener.java:388)
at weblogic.jms.client.JMSSession.onMessage(JMSSession.java:4659)
at weblogic.jms.client.JMSSession.execute(JMSSession.java:4345)
at weblogic.jms.client.JMSSession.executeMessage(JMSSession.java:3821)
at weblogic.jms.client.JMSSession.access$000(JMSSession.java:115)
at weblogic.jms.client.JMSSession$UseForRunnable.run(JMSSession.java:5170)
at
weblogic.work.SelfTuningWorkManagerImpl$WorkAdapterImpl.run(SelfTuningWorkManagerI
mpl.java:545)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
Caused By: javax.net.ssl.SSLHandshakeException:
sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target
at com.sun.net.ssl.internal.ssl.Alerts.getSSLException(Alerts.java:174)
at com.sun.net.ssl.internal.ssl.SSLSocketImpl.fatal(SSLSocketImpl.java:1731)
at com.sun.net.ssl.internal.ssl.Handshaker.fatalSE(Handshaker.java:241)
at com.sun.net.ssl.internal.ssl.Handshaker.fatalSE(Handshaker.java:235)
at
com.sun.net.ssl.internal.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.j
ava:1206)
at
com.sun.net.ssl.internal.ssl.ClientHandshaker.processMessage(ClientHandshaker.java
:136)
at com.sun.net.ssl.internal.ssl.Handshaker.processLoop(Handshaker.java:593)
at com.sun.net.ssl.internal.ssl.Handshaker.process_record(Handshaker.java:529)
at com.sun.net.ssl.internal.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:925)
```

```
at
com.sun.net.ssl.internal.ssl.SSLSocketImpl.performInitialHandshake(SSLSocketImpl.j
ava:1170)
at
com.sun.net.ssl.internal.ssl.SSLSocketImpl.readDataRecord(SSLSocketImpl.java:785)
at com.sun.net.ssl.internal.ssl.AppInputStream.read(AppInputStream.java:75)
at com.sun.mail.util.TraceInputStream.read(TraceInputStream.java:110)
at java.io.BufferedInputStream.fill(BufferedInputStream.java:218)
at java.io.BufferedInputStream.read(BufferedInputStream.java:237)
at com.sun.mail.util.LineInputStream.readLine(LineInputStream.java:88)
at com.sun.mail.smtp.SMTPTransport.readServerResponse(SMTPTransport.java:1589)
at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1369)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:412)
at javax.mail.Service.connect(Service.java:310)
at javax.mail.Service.connect(Service.java:169)
at javax.mail.Service.connect(Service.java:118)
at
oracle.sdpinternal.messaging.driver.email.EmailDriver.send(EmailDriver.java:780)
at
oracle.sdpinternal.messaging.driver.email.EmailManagedConnection.send(EmailManaged
Connection.java:50)
at
oracle.sdpinternal.messaging.driver.DriverConnectionImpl.send(DriverConnectionImpl
.java:41)
at
oracle.sdpinternal.messaging.dispatcher.DriverDispatcherBean.onMessage(DriverDispa
tcherBean.java:296)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
at
com.bea.core.repackaged.springframework.aop.support.AopUtils.invokeJoinpointUsingR
eflection(AopUtils.java:310)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.i
nvokeJoinpoint(ReflectiveMethodInvocation.java:182)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:149)
at
com.bea.core.repackaged.springframework.aop.interceptor.ExposeInvocationIntercepto
r.invoke(ExposeInvocationInterceptor.java:89)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.doProceed(DelegatingIntroductionInterceptor.java:131)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.invoke(DelegatingIntroductionInterceptor.java:119)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.framework.JdkDynamicAopProxy.invoke(Jd
kDynamicAopProxy.java:204)
```

```
at $Proxy349.onMessage(Unknown Source)
at weblogic.ejb.container.internal.MDListener.execute(MDListener.java:583)
at
weblogic.ejb.container.internal.MDListener.transactionalOnMessage(MDListener.java:
486)
at weblogic.ejb.container.internal.MDListener.onMessage(MDListener.java:388)
at weblogic.jms.client.JMSSession.onMessage(JMSSession.java:4659)
at weblogic.jms.client.JMSSession.execute(JMSSession.java:4345)
at weblogic.jms.client.JMSSession.executeMessage(JMSSession.java:3821)
at weblogic.jms.client.JMSSession.access$000(JMSSession.java:115)
at weblogic.jms.client.JMSSession$UseForRunnable.run(JMSSession.java:5170)
at
weblogic.work.SelfTuningWorkManagerImpl$WorkAdapterImpl.run(SelfTuningWorkManagerI
mpl.java:545)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
Caused By: sun.security.validator.ValidatorException: PKIX path building failed:
sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid
certification path to requested target
at sun.security.validator.PKIXValidator.doBuild(PKIXValidator.java:323)
at sun.security.validator.PKIXValidator.engineValidate(PKIXValidator.java:217)
at sun.security.validator.Validator.validate(Validator.java:218)
at
com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.validate(X509TrustManagerImpl.ja
va:126)
at
com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustMana
gerImpl.java:209)
at
com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustMana
gerImpl.java:249)
at
com.sun.net.ssl.internal.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.j
ava:1185)
at
com.sun.net.ssl.internal.ssl.ClientHandshaker.processMessage(ClientHandshaker.java
:136)
at com.sun.net.ssl.internal.ssl.Handshaker.processLoop(Handshaker.java:593)
at com.sun.net.ssl.internal.ssl.Handshaker.process_record(Handshaker.java:529)
at com.sun.net.ssl.internal.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:925)
at
com.sun.net.ssl.internal.ssl.SSLSocketImpl.performInitialHandshake(SSLSocketImpl.j
ava:1170)
at
com.sun.net.ssl.internal.ssl.SSLSocketImpl.readDataRecord(SSLSocketImpl.java:785)
at com.sun.net.ssl.internal.ssl.AppInputStream.read(AppInputStream.java:75)
at com.sun.mail.util.TraceInputStream.read(TraceInputStream.java:110)
at java.io.BufferedInputStream.fill(BufferedInputStream.java:218)
at java.io.BufferedInputStream.read(BufferedInputStream.java:237)
at com.sun.mail.util.LineInputStream.readLine(LineInputStream.java:88)
at com.sun.mail.smtp.SMTPTransport.readServerResponse(SMTPTransport.java:1589)
at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1369)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:412)
at javax.mail.Service.connect(Service.java:310)
at javax.mail.Service.connect(Service.java:169)
at javax.mail.Service.connect(Service.java:118)
at
oracle.sdpinternal.messaging.driver.email.EmailDriver.send(EmailDriver.java:780)
at
oracle.sdpinternal.messaging.driver.email.EmailManagedConnection.send(EmailManaged
```

```
Connection.java:50)
at
oracle.sdpinternal.messaging.driver.DriverConnectionImpl.send(DriverConnectionImpl
.java:41)
at
oracle.sdpinternal.messaging.dispatcher.DriverDispatcherBean.onMessage(DriverDispa
tcherBean.java:296)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
at
com.bea.core.repackaged.springframework.aop.support.AopUtils.invokeJoinpointUsingR
eflection(AopUtils.java:310)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.i
nvokeJoinpoint(ReflectiveMethodInvocation.java:182)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:149)
at
com.bea.core.repackaged.springframework.aop.interceptor.ExposeInvocationIntercepto
r.invoke(ExposeInvocationInterceptor.java:89)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.doProceed(DelegatingIntroductionInterceptor.java:131)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.invoke(DelegatingIntroductionInterceptor.java:119)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.framework.JdkDynamicAopProxy.invoke(Jd
kDynamicAopProxy.java:204)
at $Proxy349.onMessage(Unknown Source)
at weblogic.ejb.container.internal.MDListener.execute(MDListener.java:583)
at
weblogic.ejb.container.internal.MDListener.transactionalOnMessage(MDListener.java:
486)
at weblogic.ejb.container.internal.MDListener.onMessage(MDListener.java:388)
at weblogic.jms.client.JMSSession.onMessage(JMSSession.java:4659)
at weblogic.jms.client.JMSSession.execute(JMSSession.java:4345)
at weblogic.jms.client.JMSSession.executeMessage(JMSSession.java:3821)
at weblogic.jms.client.JMSSession.access$000(JMSSession.java:115)
at weblogic.jms.client.JMSSession$UseForRunnable.run(JMSSession.java:5170)
at
weblogic.work.SelfTuningWorkManagerImpl$WorkAdapterImpl.run(SelfTuningWorkManagerI
mpl.java:545)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
Caused By: sun.security.provider.certpath.SunCertPathBuilderException: unable to
find valid certification path to requested target
at
sun.security.provider.certpath.SunCertPathBuilder.engineBuild(SunCertPathBuilder.j
```

```
ava:174)
at java.security.cert.CertPathBuilder.build(CertPathBuilder.java:238)
at sun.security.validator.PKIXValidator.doBuild(PKIXValidator.java:318)
at sun.security.validator.PKIXValidator.engineValidate(PKIXValidator.java:217)
at sun.security.validator.Validator.validate(Validator.java:218)
at
com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.validate(X509TrustManagerImpl.ja
va:126)
at
com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustMana
gerImpl.java:209)
at
com.sun.net.ssl.internal.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustMana
gerImpl.java:249)
at
com.sun.net.ssl.internal.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.j
ava:1185)
at
com.sun.net.ssl.internal.ssl.ClientHandshaker.processMessage(ClientHandshaker.java
:136)
at com.sun.net.ssl.internal.ssl.Handshaker.processLoop(Handshaker.java:593)
at com.sun.net.ssl.internal.ssl.Handshaker.process_record(Handshaker.java:529)
at com.sun.net.ssl.internal.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:925)
at
com.sun.net.ssl.internal.ssl.SSLSocketImpl.performInitialHandshake(SSLSocketImpl.j
ava:1170)
at
com.sun.net.ssl.internal.ssl.SSLSocketImpl.readDataRecord(SSLSocketImpl.java:785)
at com.sun.net.ssl.internal.ssl.AppInputStream.read(AppInputStream.java:75)
at com.sun.mail.util.TraceInputStream.read(TraceInputStream.java:110)
at java.io.BufferedInputStream.fill(BufferedInputStream.java:218)
at java.io.BufferedInputStream.read(BufferedInputStream.java:237)
at com.sun.mail.util.LineInputStream.readLine(LineInputStream.java:88)
at com.sun.mail.smtp.SMTPTransport.readServerResponse(SMTPTransport.java:1589)
at com.sun.mail.smtp.SMTPTransport.openServer(SMTPTransport.java:1369)
at com.sun.mail.smtp.SMTPTransport.protocolConnect(SMTPTransport.java:412)
at javax.mail.Service.connect(Service.java:310)
at javax.mail.Service.connect(Service.java:169)
at javax.mail.Service.connect(Service.java:118)
at
oracle.sdpinternal.messaging.driver.email.EmailDriver.send(EmailDriver.java:780)
at
oracle.sdpinternal.messaging.driver.email.EmailManagedConnection.send(EmailManaged
Connection.java:50)
at
oracle.sdpinternal.messaging.driver.DriverConnectionImpl.send(DriverConnectionImpl
.java:41)
at
oracle.sdpinternal.messaging.dispatcher.DriverDispatcherBean.onMessage(DriverDispa
tcherBean.java:296)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:
25)
at java.lang.reflect.Method.invoke(Method.java:597)
at
com.bea.core.repackaged.springframework.aop.support.AopUtils.invokeJoinpointUsingR
eflection(AopUtils.java:310)
at
```

```
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.i
nvokeJoinpoint(ReflectiveMethodInvocation.java:182)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:149)
at
com.bea.core.repackaged.springframework.aop.interceptor.ExposeInvocationIntercepto
r.invoke(ExposeInvocationInterceptor.java:89)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.doProceed(DelegatingIntroductionInterceptor.java:131)
at
com.bea.core.repackaged.springframework.aop.support.DelegatingIntroductionIntercep
tor.invoke(DelegatingIntroductionInterceptor.java:119)
at
com.bea.core.repackaged.springframework.aop.framework.ReflectiveMethodInvocation.p
roceed(ReflectiveMethodInvocation.java:171)
at
com.bea.core.repackaged.springframework.aop.framework.JdkDynamicAopProxy.invoke(Jd
kDynamicAopProxy.java:204)
at $Proxy349.onMessage(Unknown Source)
at weblogic.ejb.container.internal.MDListener.execute(MDListener.java:583)
at
weblogic.ejb.container.internal.MDListener.transactionalOnMessage(MDListener.java:
486)
at weblogic.ejb.container.internal.MDListener.onMessage(MDListener.java:388)
at weblogic.jms.client.JMSSession.onMessage(JMSSession.java:4659)
at weblogic.jms.client.JMSSession.execute(JMSSession.java:4345)
at weblogic.jms.client.JMSSession.executeMessage(JMSSession.java:3821)
at weblogic.jms.client.JMSSession.access$000(JMSSession.java:115)
at weblogic.jms.client.JMSSession$UseForRunnable.run(JMSSession.java:5170)
at
weblogic.work.SelfTuningWorkManagerImpl$WorkAdapterImpl.run(SelfTuningWorkManagerI
mpl.java:545)
at weblogic.work.ExecuteThread.execute(ExecuteThread.java:256)
at weblogic.work.ExecuteThread.run(ExecuteThread.java:221)
>
```

Solution

Ensure the following:

- Username and password provided are correct.
- Entry for DemoTrust.jks is removed from setDomainEnv script.
- Application server, such as Oracle WebLogic Server, has been deployed and configured correctly.
- Certificate exchange is done.

19.6.5 Issues Related to Failed Email Delivery Not Reported Through EM

Problem

Status Code is always DELIVERY_TO_GATEWAY_SUCCESS in Enterprise Manager (EM) Usermessagingserver Message Status, although the email is invalid. The status code does not update to failure even if the user does not receive any email.

Solution

Ensure that the following Incoming settings in the Driver configuration are properly configured:

- MailAccessProtocol
- ReceiveFolder
- IncomingMailServer
- IncomingMailServerPort
- IncomingMailServerSSL
- IncomingMailIDs
- IncomingUserIDs
- IncomingUserPasswords
- ImapAuthPlainDisable

For additional information on , see "Configuring Oracle User Messaging Service" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

20

Configuring Oracle Identity Manager

This chapter describes how to configure Oracle Identity Manager deployment by using system configuration properties. It contains the following sections:

- Managing System Properties
- Configuring Oracle Identity Manager Components
- Configuring the Access Catalog
- Configuring the Identity Provider

20.1 Managing 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 which you can control the configuration of Oracle Identity Manager.

This section describes the following topics:

- System Properties in Oracle Identity Manager
- Creating and Managing System Properties

20.1.1 System Properties in Oracle Identity Manager

Table 20–1 lists and describes the default system properties in Oracle Identity Manager.

Table 20–1 Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Access Policy Revoke If No Longer Applies Enhancement	XL.AccessPolicyRevokeIfNoLongerAppliesEnhancement	FALSE	Determines if the Revoke if no longer applies flag in access policy is applicable. If the value is true, then this flag is applicable to child table data (entitlements) along with parent data. The user can determine if child data must be removed or retained when access policy no longer applies to user based on this flag. If the value if false, then child table data (entitlements) are always removed after access policy is no longer applied. **Note**: This property is not used in Oracle Identity Manager Release 2 (11.1.2) or later.
Allows access policy based provisioning of multiple instances of a resource	XL.AllowAPBasedMultipleAccountProvisioning	FALSE	Determines if multiple instances of a resource can be provisioned to multiple target resources. When the value is false, provisioning multiple instances of resource object via access policy is not allowed. When the value is true, provisioning multiple instances of resource object via access policy is allowed.
Allows control over role hierarchical access policy evaluation	XL.AllowRoleHierarchicalPolicyEval	FALSE	This property is used to control allowing role hierarchical access policy evaluation. When this system property is set to TRUE, access from inherited access policies is given to the user. If set to FALSE, access from access policies attached to inherited roles is not given to the user.
Allows linking of access policies to reconciled and bulk loaded accounts	XL.AllowAPHarvesting	FALSE	Determines if access policy engine can link access policies to reconciled accounts and to accounts created by the Bulk Load Utility. This property is used in the context of evaluating access policies for reconciled accounts and to accounts created by the Bulk Load Utility. For more information, see "Evaluating Policies for Reconciled and Bulk Load-Created Accounts" on page 5-5. **Note:** This property is used in Oracle Identity Manager 11*g* Release 2 (11.1.2.2.0) or later.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Are challenge questions disabled in OIM	OIM.DisableChallengeQuestions	FALSE	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. When the value is FALSE, then PWR.PWR_CHA_POLICY_ENABLED is honored to determine if challenge policy is enabled or not.
Catalog Additional Application Details Task Flow	CatalogAdditionalApplicationDetailsTaskFlow	/WEB-INF/oracle/iam/ui/common/tfs/empty-tf.xml#empty-tf	A custom task flow is to be displayed when an application is selected from the catalog checkout page. The task flow page will display as a tab in the cart details section.
Catalog Additional Entitlement Details Task Flow	CatalogAdditionalEntitlementDetailsTaskFlow	/WEB-INF/oracle/iam/ui/common/tfs/empty-tf.xml#empty-tf	A custom task flow is to be displayed when an entitlement is selected from the catalog checkout page. The task flow page will display as a tab in the cart details section.
Catalog Additional Role Details Task Flow	CatalogAdditionalRoleDetailsTaskFlow	/WEB-INF/oracle/iam/ui/common/tfs/empty-tf.xml#empty-tfs	A custom task flow is to be displayed when a role item is selected from the catalog checkout page. The task flow page will display as a tab in the cart details section.
Catalog Advanced Search Maximum Applications	CatalogAdvancedSearchMaxApps	15	In the default form for catalog advanced search, you can search for entitlements by specifying the list of applications to search from. This system property controls the maximum number of applications that can be selected for entitlement search.
Catalog Advanced Search Taskflow	CatalogAdvancedSearchTaskflow	/WEB-INF/oracle/iam/ui/catalog/tfs/catalog-advanced-search-tf.xml#catalog-advanced-search-tf	Determines the taskflow used for catalog search. If you create custom taskflow for catalog search, then change the value of this property to the complete path of the custom taskflow.
Catalog Attributes for Sorting Search Results	CatalogSortAttributes	ENTITY_DISPLAY_NAME; ENTITY_TYPE	This property determines the attributes that are displayed in the Sort By drop down in the catalog results tab.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Catalog Audit Data Collection	XL.CatalogAuditDataCollection	none	Determines if catalog auditing is enabled or disabled. The default value is `none`, which specifies that catalog auditing in disabled. To enable catalog auditing, set the value of this property to `catalog`.
Category count option can be 0, 1 or 2	CATALOG.CATEGORY_COUNT_OPTION	2	Determines what is displayed in the category count block. If the value is 0, then category count block is deactivated. If the value is 1, then distinct categories across the system are displayed without respective category count. If the value is 2, then categories with count are displayed. **Note:** It is recommended that the values be modified if poor catalog search performance is experienced. **Note:** This system property is available only after you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170718. For information about downloading and applying the bundle patch, refer to bundle patch documentation.
Catalog Regex for special characters	Catalog.SpecialCharacterRegex	[^\w]	Enables text parsing and escaping of special characters when performing a catalog search by using some special characters. If you do not want any text parsing and escaping of special characters, then change the value of this property to `[^\w^\W]`.
Catalog search MAX result size. Default value is -1 which means return all	XL.CatalogSearchResultCap	-1	When the data is huge in the request catalog and you encounter any issue with the performance of the catalog, you can change the value of this system property and provide some reasonable values, such as 500. As a result, catalog search will not return more than the specified value. If the value is -1, then no result size limit is applied on the catalog search result.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Catalog Searchable UDF In Tags	CATALOG.SearchableUdfInTags	FALSE	If want to use searchable UDF in TAGS, then you can set the value of this property to TRUE. Then, you can run the scheduled task in recalculate tags mode and searchable UDF values are part of the TAGS column. The same value can be used in keyword search.
Catalog Table Rows To Display Size	CatalogTableRowsToDisplaySize	10	This property is used to control the number of rows displayed in all tables found in all catalog-related pages. **Note:** The value of this system property must be less than or equal to 50.
CommonName generation plugin	XL.DefaultCommonNamePolicyImpl	oracle.iam.ldapsync.impl.plugins.FirstNameLastNamePolicy	Determines the common name generation plugin to generate common name.
Compiler Path for Connectors	XL.CompilerPath		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.
Compute and Persist Min Age On Password Change	ComputePersistMinAgeOnPasswordChange	proactive	Password minimum age calculation has two modes, proactive and reactive mode. In proactive, where minimum age date is calculated at password change time, any subsequent change to the user's applicable password policy's minimum age property will not be honored until the next password change, where as with the reactive approach, policy changes are applied immediately. To enable proactive or reactive approach, system property Compute Persist Min Age On Password Change is introduced.
Copy both user and manager of user in the create user email notification	XL.NotifyUserCreateToOther	TRUE	Copies the user and user's manager in the email notification that is sent when a user is created.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Data Collection Session ID	XL.DataCollectionSessionID	dummy	Specifies the session ID of the current Oracle Identity Analytics (OIA) Data collection session.
Data Collection Status	XL.DataCollectionStatus	FINALIZED	Specifies the status of the current OIA data collection session.
Default Date Format	XL.DefaultDateFormat	yyyy/mm/dd hh:mm:ss z	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.
Default policy for username generation	XL.DefaultUserNamePolicyImpl	oracle.iam.identity.usermgmt.impl.plugins.DefaultComboPolicy	Determines the username policy to use when generating a username.
Default user name domain	XL.UserNameDomain	oracle.com	This property is used by the DefaultComboPolicy to generate a user name in e-mail format.
Disable Catalog Blank Search	CATALOG.DISABLE_BLANK_SEARCH	TRUE	This property is used to enable or disable blank text search in the catalog. If the value is TURE, then blank text search is disabled. If the value is FALSE, then blank text search is enabled. **Note:** It is recommended that the values be modified if poor catalog search performance is experienced. **Note:** This system property is available only after you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170718. For information about downloading and applying the bundle patch, refer to bundle patch documentation.
Disabling Default Search of UI pages	OIG.DisableDefaultTableSearches	FALSE	This property is used to enable or disable blank text search in the Users, Roles, Organizations, and Administration Roles page. If the value is TRUE, then blank text search is enabled. If the value is FALSE, then blank text search is disabled. **Note**: This system property is available only after you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.180331.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Display Certification or Attestation	OIM.ShowCertificationOrAttestation	attestation	This property has been superseded by the `Identity Auditor Features Enabled` system property, and attestation is no longer supported. **Note:** In this release, this property is not used as Attestation is not supported. This property is superceded by the `Identity Auditor Features Enabled` system property.
Does user have to provide challenge information during registration	PCQ.PROVIDE_DURING_SELFREG	TRUE	If the value is TRUE, then users will have to provide challenge information during registration.
Email Server	XL.MailServer	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.
Email Validation Pattern	XL.EmailValidationPattern	[A-Za-z0-9\._\#\!\$\&\'*\/\=\?\^\`\{\}\~|\%\+\-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}	This property contains the regular expression used to validate the email ID of a user.
Enable disabled resource instances when a user is enabled	XL.EnableDisabledResources	TRUE	If the value is TRUE, then the disabled resource instances are enabled when a user is enabled.
Enable Exception Reports	XL.EnableExceptionReports	TRUE	This property is used to enable the exception reporting feature. Exception reporting is enabled only if the value is set to TRUE.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Evaluate LDAP Container Rules for Entity Modification	LDAPEvaluateContainerRulesForModify	FALSE	If the property value is TRUE, then the LDAP container rules defined in LDAPContainerRules.xml are evaluated for entity modification. However, if none of the rules match, then the default container is not returned. The original parent container of the entity is returned, which means that there is no change in the entity DN. If the property value is FALSE, then the LDAP container rules defined in LDAPContainerRules.xml are not evaluated. The entity DN does not change. **Note:** This property only applies to a modification scenario and not to the entity creation scenario.
Force to set questions at startup	PCQ.FORCE_SET_QUES	False	When the user logs into the Oracle Identity Self Service or Oracle Identity System Administration for the first time, the user must set the default questions for resetting the password. **Note:** After modifying the value of this property, you must restart Oracle Identity Manager server for the changes to take effect.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
GTC Auto Import	XL.GTCAutoImport	true	Based on the value of this property, the DM xml that is generated while Generic Technology Connector (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: *OIM_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
Homepage for Self Service console	OIM.IdentityHomepage	none	This property is used to set the page to be displayed after a user logs in to Oracle Identity Manager Self Service. You can set one of the following as the value of this property: ■ **my_access:** Displays the My Access page ■ **my_info:** Displays the My Information page ■ **home:** Displays the Home page ■ **catalog_home:**Displays the Catalog page ■ **none:** Displays no page After modifying the value of this property, you must restart Oracle Identity Manager server for the changes to take effect. **Note:** This property is not used in Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0).

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Identity Auditor Feature Set Availability	OIG.IsIdentityAuditorEnabled	FALSE	When the value of this property is TRUE, role lifecycle management, Segregation of Duties (SoD), and identity certification are enabled. **Note:** After modifying the value of this system property, you must restart Oracle Identity Manager server for the changes to take effect.
Inbox Task Tabs (none/all)	UI.INBOX.VIEW.TaskTabs	none	This property determines whether or not to show additional links, such as Initiated tasks, Reportees and Administrative tasks, in the Inbox. When set to all, the following links are displaied in the Inbox: My tasks,Initiated tasks, Reportees, Administrative tasks. When set to none, only the My tasks link is displayed in the Inbox.
Indicates if referential integrity is enabled in target LDAP directory	XL.IsReferentialIntegrityEnabledInLDAP	FALSE	The value of this property is TRUE if referential integrity in target LDAP directory is turned on. The value of this property is FALSE if referential integrity in target LDAP directory is turned off. To be able to modify an entity stored in LDAP, this prop must be set to TRUE.
Is DataProvider LDAP/DB	OIM.DataProvider	DB	Specifies the data provider, which is Oracle Identity Manager database. The default value is DB, which indicates that the database is the data provider.
Is disabled manager allowed	AllowDisabledManagers	FALSE	Specifies whether a user in the disabled state can be set as a manager for another user.
Is OIM Notifications disabled (true/false)	XL.DisableAllNotifications	false	This property is used to enable or disable all notifications in Oracle Identity Manager. When the value of this property is set to false, notifications are enabled. When the value of this property is true, notifications are disabled.
Is Self-Registration Allowed	XL.SelfRegistrationAllowed	TRUE	If the value is TRUE, then the users are allowed to self-register.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
LDAP Reservation Plugin	XL.LDAPReservationPluginImpl	oracle.iam.identity.usermgmt.impl.plugins.reservation.ReservationInOID	This property determines the LDAP reservation plugin implementation to be picked up for reservation of user attributes.
Maximum number of records to be fetched from Catalog	Catalog.SearchResultCap	-1	This property determines how many records must be fetched from the catalog when a search is performed. If the value is -1, then all records are fetched from the catalog table. If the value is 10000, then only 10000 records are fetched from the catalog. **Note:** It is recommended that you set the value to 10000 only if poor catalog search performance is experienced. **Note:** This system property is available only after you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170718. For information about downloading and applying the bundle patch, refer to bundle patch documentation.
Level of Role Auditing	XL.RoleAuditLevel	None	This property controls the amount of audit data collected when an operation is performed on a role, such as creation or modification. The supported levels are: ■ **None:** No audit data is collected. ■ **Role:** Creation, modification, and deletion of role is audited. ■ **Role Hierarchy:** Changes made to the role inheritance is audited.
Notify other recipients with the password reset email if email of user is null	XL.NotifyPasswordGenerationToOther	TRUE	When the value of this property is TRUE, the email notification for reset password is sent to other recipients if the email ID of the user is not specified.
Number of records to be executed in a batch during Catalog Enrichment	XL.CatalogEnrichmentBatchSize	500	This property determines how many records must be processed in a batch by the catalog job during catalog enrichment.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
OIA integration status	OIM.IsOIAIntegrationEnabled	FALSE	Specifies whether OIA is integrated with Oracle Identity Manager. Set the value of this property to `TRUE` before you add role memberships in Oracle Identity Manager. If you set the value of this property to `FALSE`, incremental role memberships into OIA will not work. **Note:** You must do a full import of role memberships at least once after this property is enabled.
Old Password Validator	OIM.OldPasswordValidator	oracle.iam.identity.usermgmt.impl.ContainerLoginPasswordVerifier	The property specifies the name of the plugin class to be used for verifying old passwords.
OMSS Enabled	OMSS Enabled	false	When the value of this property is true, OMSS integration is enabled, and the OMSS links and tabs are displayed in Oracle Identity Self Service. **Note:** After modifying the value of this system property, you must restart Oracle Identity Manager server for the changes to take effect.
Period to Delay User Delete	XL.UserDeleteDelayPeriod	0	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. If this property is not set, then the user is automatically deleted at the expiration of the end date by the Disable/Delete User After End Date scheduled job.
Proxy User Email Notification	XL.ProxyNotificationTemplate	Notify Proxy User	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 the proxy for some other user.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Recon Batch Size	OIM.ReconBatchSize	500	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. **Note:** When using trusted source reconciliation from Oracle Directory Server Enterprise Edition (ODSEE), the value of this property must not be 0. When the value is 0, users are not created in Oracle Identity Manager. **Note:** You must restart Oracle Identity Manager server after setting this property.
Request Notification Level	RequestNotificationLevel	0	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. This property can have the following values: ■ **0:** The notification feature is disabled. ■ **1:** Notifications are sent for every change in request status. ■ **2:** Notifications are sent for request creation and change of status to any of the Request End statuses. Request End statuses include Request Failed and other failure related statuses, Request Completed, Request Withdrawn, and Request Closed. ■ **3:** Email notifications are sent only on request completion. For request notification level 2, notifications are sent for request creation and change of status to any of the Request End statuses. Request End statuses include Request Failed and other failure related statuses, Request Completed, Request Withdrawn, and Request Closed.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Retry Count for recon event	Recon.RetryCount	5	This property determines the reconciliation retry count. The retry count value is picked up from the value of this property. If you specify a value that is greater than 0, then auto retry is configured. If you specify 0 as the value of this property, then auto retry is not configured.
Search Stop Count	XL.IDADMIN_STOP_COUNT	300	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.
Segregation of Duties (SOD) Check Required	XL.SoDCheckRequired	FALSE	This property indicates whether or not Segregation of Duties (SoD) check is required.
Send email notification based on user locale	XL.SendEmailNotificationBasedOnUserLocale	false	This property determines whether an email notification is sent based on the receiver's (user/manager/assignee/requestor) locale when the value is set to true. If the value is set to false, then notification is sent in the server locale. **Note:** This system property has been deprecated in this release of Oracle Identity Manager.
Should send notifications in recon or not	Recon.SEND_NOTIFICATION	true	Determines if notification is sent to the user when the user login and password are generated in postprocess event handler for user creation via trusted source reconciliation. If the value is set to true, then notification is sent when user login and password are generated in postprocess event handler for user creation via trusted source reconciliation. If the value is set to false, then notification is not sent when user login and password are generated in postprocess event handler for user creation via trusted source reconciliation.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Shows tasks assigned to group users with highest priority or least load only	XL.ShowTaskAssignedToGroupUserOnly	FALSE	If the value is TRUE, then the tasks are assigned to group users with highest priority or least load only when the assignment type is Group User With Least Load.
Specifies the LDAP container mapper plug-in to be used	LDAPContainerMapperPlugin	oracle.iam.ldapsync.impl.DefaultLDAPContainerMapper	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.
URL for challenge questions modification	OIM.ChallengeQuestionsModificationURL	NONE	When a user is locked, an automatic unlock occurs after a prescribed time period. This property defines that time period in seconds. Therefore, for example, if a user account is locked and the value of this property is 86400 seconds (one day), then the account is automatically unlocked after one day. 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
URL for change password	OIM.ChangePasswordURL	NONE	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
User Attribute Reservation Enabled	XL.IsUsrAttribReservEnabled	TRUE	This property is used to enable user attribute reservation.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
User Id reuse property.Requires dropping the index present on USR_LOGIN column	XL.UserIDReuse	FALSE	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. **Note:** It is imperative to de-provision all accounts associated with a deleted user, because if you create a new user with the same user name as that of the deleted user by setting the `XL.UserIDReuse` property to `true`, then the new user might get access to offline accounts of the deleted user that was not deleted as part of the de-provisioning process.
User Language	user.language	en	The user.language value is configured during installation for Locale handling at server side.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
User profile audit data collection level	XL.UserProfileAuditDataCollection	Resource Form	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.
User Region	user.region	US	The user.region value is configured during installation for Locale handling at server side.
Whether or not email should be validated for uniqueness	OIM.EmailUniqueCheck	TRUE	This property is available in an Oracle Identity Manager 11*g* Release 2 (11.1.2.1.0) deployment that has been upgraded from an earlier release of Oracle Identity Manager. If the value of this property is FALSE, then Email Uniqueness check is not performed by Oracle Identity Manager. If the value if TRUE, then Email Uniqueness check is performed by Oracle Identity Manager. **Note:** If this property is not present, then Email Uniqueness check is performed by Oracle Identity Manager.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
Workflows Enabled	Workflows Enabled	TRUE	This property determines whether SOA server is turned on or turned off. If the value of this property is TRUE, then SOA sever is turned on. If the value of this property is FALSE, then SOA server is turned off. **Note:** After setting the value of this system property, you must restart Oracle Identity Manager. **Note:** Toggling between enabling and disabling workflows is not supported.
Workflow Policies Enabled	Workflow Policies Enabled	TRUE	This property determines whether approval workflows is enabled or disabled in Oracle Identity Manager. Approval workflows is used to determine if operation requires approval or not, and if approval is required, then which workflow is to be invoked. If the value of this property is TRUE, then approval workflow is enabled. If the value of this property is FALSE, then approval workflow is disabled. For detailed information about approval workflow, see Chapter 4, "Managing Workflows".
UserPostProcessActionHandlerUsingAsync	XL.UserPostProcessActionHandlerUsingAsync	false	If you set the value to true, then the user membership evaluation on user create/update takes place asynchronously. If the value is false, then the user membership evaluation and subsequent role grant/revoke takes place in sync and can be a time-consuming operation. If there are large number of rules and roles and create/update operation takes a few seconds, then the recommendation is to set the value of this property to true. **Note:** This system property is available only if you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170418. For instructions on downloading and applying the bundle patch, refer to bundle patch documentation.

Table 20–1 (Cont.) Default System Properties in Oracle Identity Manager

Property Name	Keyword	Default Value	Description
TriggerUserProcessesUsingAsync	XL.TriggerUserProcessesUsingAsync	false	When you update a user, the change is propagated to all provisioned accounts. For a large number of accounts, this can cause significant delay. Setting this property to true updates the accounts asynchronously. **Note:** This system property is available only if you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170418. For instructions on downloading and applying the bundle patch, refer to bundle patch documentation.
SelfServiceNotificationUsingAsync	XL.SelfServiceNotificationUsingAsync	false	After a user account is created or password is reset, an email notification is sent synchronously. Setting the value of this property to true sends out the notification asynchronously. **Note:** This system property is available only if you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170418. For instructions on downloading and applying the bundle patch, refer to bundle patch documentation.
ExecuteDymanicRoleMembershipOrchUsingAsync	XL.ExecuteDymanicRoleMembershipOrchUsingAsync	false	If XL.UserPostProcessActionHandlerUsingAsync is false, then the user membership rule evaluation takes place in sync. But you can still have the role grant/revoke happen asynchronously by setting the value of this property to true. **Note:** This system property is available only if you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170418. For instructions on downloading and applying the bundle patch, refer to bundle patch documentation.

Note: In this release of Oracle Identity Manager, the `XL.MAXLOGINATTEMPTS` and `XL.MAXPASSWORDRESETATTEMPTS` system properties have been removed.

The function of the `XL.MAXLOGINATTEMPTS` system property has been replaced with the `Maximum Incorrect Login attempts counter` field in the password policy details page. The function of the `XL.MAXPASSWORDRESETATTEMPTS` system property has been replaced with the `Lock User After Attempts` field in the Challenge Options section of the password policy details page. For information about these fields, see "Managing Password Policies" in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager*.

Oracle Identity Manager provides a set of system properties that are not present in the PTY table by default. You can add these system properties to the PTY table by using the Oracle Identity System Administration, and then use the properties to change some of the default settings in Oracle Identity Manager. For example, if you want to configure the number of times Oracle Identity Manager retries to get a connection when the JDBC connection fails, then you can configure the JDBC Connection Retry Attempts system property.

> **See Also:** *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for more information about GTC

Table 20–2 lists the system properties you can add to the PTY table:

Table 20–2 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
OIM Paging Limit	Default paging limit for search operations on user entity.	OIM.PagingLimit	300
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

Table 20–2 (Cont.) Nondefault System Properties

Property Name	Description	Keyword	Sample Value
Allowed Back URLs	This property is required if you want to setup any non-OIM/OAM URLs to be a valid backURL on the Track Self Registration Request page. Oracle Identity Manager validates the back URLs and redirect URLs against a list of URLs provided by this system property. The value of this property is a comma-separated list of URLs that Oracle Identity Manager allows for redirection.	XL.AllowedBackURLs	http://*OIM_HOST*:*OIM_PORT*/
Allowed Back URLs Mode	This system property determines the mode in which the XL.AllowedBackURLs system property works. It has the following possible values: ■ **Enforce:** Ensure that the current URL is present in the white list specified as the value of XL.AllowedBackURLs. If not present, then change the back URL to the default URL, which is the sign-in page. ■ **Disable:** Log all the white list validations. The default value is `Enforce`.	XL.AllowedBackURLsMode	Enforce

Table 20–2 (Cont.) Nondefault System Properties

Property Name	Description	Keyword	Sample Value
XL.AllowedOrigins	Allows users to set the whitelist for the CORS filter. **Note:** This property is not automatically added after applying Bundle Patch 11.1.2.3.161018. Therefore, you must add this system property manually. If you do not set this property, then the CORS filter will not allow CORS request.	XL.AllowedOrigins	Apply the following guidelines for specifying the value: ■ The URLs can be comma separated, for example http://www.example.com:14001 and https://www.test.com:14003. ■ The URL can contain simple wildcard matching (for example http://*.example.com:14000, http://*.com:14001) with only a single '*' character. '*.example.*.com' will not work correctly. ■ The pattern matching is very simple and only pertains to the domain part of the URL. No matching on scheme or port is supported (*://example.com:14001 or http://example.com:*). ■ Only http and https schemes are supported. ■ The matching goes from right to left. ■ The '*' will only match the text of domain after the period and before the next period. Patterns such as '*ampl.com' and '*.*partial*.c*' are not supported. ■ A single '*' will match anything and should be used for test/development only.

20.1.2 Creating and Managing System Properties

This section discusses the following topics:

- Searching for System Properties
- Modifying System Properties

- Purging Cache

20.1.2.1 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

20.1.2.1.1 Performing a Simple Search To perform a simple search for system properties:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Configuration Properties**.
3. 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 search without any value or with wild card character * 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.
4. Click the icon next to the Search field. A list of all system properties that meet the search criterion is displayed.

   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.

20.1.2.1.2 Performing an Advanced Search To perform an advanced search for system properties:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. In the left pane of the System Configuration section, click **Advanced Search**. The Properties: Advanced Search page is displayed.
3. In the list adjacent to the Property Name field, select a search condition.
4. 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 Begins with, Contains, Does not begin with, Does not contain, Does not end with, Does not equal, Ends with, Equals, Is not present, and Is present.
5. Click **Search**. The system properties that match the search criterion are displayed in the search results table.

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

20.1.2.2 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.

You cannot modify the Property Name and Keyword fields of a system property created in a non-English locale. As a workaround, delete the existing system property and create a new one with the desired values.

In an English locale, non-ASCII characters are allowed in a system property name. When you modify the name of a system property to include non-ASCII characters, you must ensure the following if you want the changes to be translated into other languages:

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.
3. Modify the values in the fields. Generally, you need to modify the Value field to change the functionality that the system property provides.
4. Click **Save** to save the changes made.

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

20.1.2.3 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:

To clear the server cache:

1. Depending on the operating system being used, navigate to the following directory:
   - For Microsoft Windows:

     OIM_HOME\server\bin\
   - For UNIX:

     OIM_HOME/server/bin/
2. Run one of the following commands:
   - For Microsoft Windows:

     ```
     PurgeCache.bat CATEGORY_NAME
     ```
   - For UNIX:

     ```
     sh PurgeCache.sh CATEGORY_NAME
     ```

   The *CATEGORY_NAME* name argument represents the Oracle Identity Manager category name that is to be purged, for example, FormDefinition.

   To purge all the categories, pass a value of "All" to the PurgeCache utility. It is recommended to clear all the categories.

   ```
   sh PurgeCache.sh All
   ```

20.2 Configuring Oracle Identity Manager Components

This section describes how to configure the following functionalities in Oracle Identity Manager:

20.2.1 Configuring Product Options

You can configure the availability of some of the features in Oracle Identity Manager with the help of system properties. To do so:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Configuration Properties**.
3. Enable role lifecycle management, Segregation of Duties (SoD), and identity certification. To do so:
   a. Search for the `Identity Auditor Feature Set Availability` system property with keyword `OIG.IsIdentityAuditorEnabled`.

      The default value of this property is `FALSE`, which means that role lifecycle management, Segregation of Duties (SoD), and identity certification are disabled by default.

   b. Modify the value of the property to `TRUE`.
   c. Click **Save**.
4. Enable the integration with Oracle Mobile Security Suite (OMSS). To do so:
   a. Search for the `OMSS Enabled` system property with keyword `OMSS Enabled`.

      The default value of this system property is `false`, which means that the OMSS integration is disabled, and the OMSS links and tabs are not displayed in Oracle Identity Self Service.

   b. Modify the value of the property to `true`.
   c. Click **Save**.
5. Enable the integration with Oracle Identity Analytics (OIA). To do so:
   a. Search for the `OIA Integration Status` system property with keyword `OIM.IsOIAIntegrationEnabled`.

      The default value of this property is `FALSE`, which means that integration with OIA is disabled by default.

   b. Modify the value of the property to `TRUE`.
   c. Click **Save**.
6. Restart Oracle Identity Manager.

   You must restart Oracle Identity Manager after modifying the values of each of the `Identity Auditor Feature Set Availability`, `OIA Integration Status`, and `OIA Integration Status` system properties.

20.2.2 Configuring the URL for Challenge Questions

To configure the URL within Oracle Adaptive Access Manager (OAAM) that handles challenge questions:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. Search for the `URL for challenge questions modification` system property with keyword `OIM.ChallengeQuestionsModificationURL`.

   The default value of this property is `NONE`.
3. Modify the value of the property to specify the URL within OAAM that handles the challenge questions. For example:

   ```
   http://OAAM_HOST:OAAM_PORT/OAAM_SERVER/userPreferences.do?showView=registerQuestions
   ```
4. Click **Save**.

20.2.3 Configuring the URL for Change Password

To configure the URL within OAAM that handles change password:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. Search for the `URL for change password` system property with keyword `OIM.ChangePasswordURL`.

   The default value of this property is `NONE`.
3. Modify the value of the property to specify the URL within OAAM that handles the change password functionality. For example:

   ```
   http://OAAM_HOST:OAAM_PORT/OAAM_SERVER/userPreferences.do?showView=changePassword
   ```
4. Click **Save**.

20.2.4 Enabling Challenge Questions

To enable challenge questions in Oracle Identity Manager:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. Challenge questions in Oracle Identity Manager are controlled by a combination of three system properties. Search and specify values for the following system properties:
   - **Are challenge questions disabled in OIM:** Determines whether 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 OAAM configuration. When the value is `TRUE`, the challenge questions are handled by OAAM.
   - **Force to set questions at startup:** Determines whether or not the user must set the default questions for resetting the password when the user logs into the Oracle Identity Self Service or Oracle Identity System Administration for the first time. When the value is `FALSE`, the user is not forced to set the default

questions for resetting the password on first login. When the value is TRUE, the user must set the default questions for resetting the password on first login.

After modifying the value of this property, you must restart Oracle Identity Manager server for the changes to take effect.

- **Does user have to provide challenge information during registration:** Determines whether or not users must provide challenge information during registration. When the value is TRUE, user must provide challenge information during registration.

3. Save the system property values.

20.2.5 Configuring Username Generation

To configure username generation:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. Username generation in Oracle Identity Manager is controlled by a combination of three system properties. Search and specify values for the following system properties:
   - **Default policy for username generation:** Determines the username policy to use when generating a username. The default value is `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. The default value is `oracle.com`.
   - **CommonName generation plugin:** Determines the common name generation plugin to generate common name. The default value is `oracle.iam.ldapsync.impl.plugins.FirstNameLastNamePolicy`.
3. Save the system property values.

20.2.6 Configuring User ID Reuse

To configure user ID reuse:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. Search for the `User Id reuse property.Requires dropping the index present on USR_LOGIN column` system property with keyword `XL.UserIDReuse`.

   This property determines whether or not a deleted user account can be reused.
3. To reuse a deleted user account, modify the value of this property to TRUE, and drop the unique index for the USR_LOGIN column in the USR table and create a non unique index. To prevent a user account from being reused, assign this property a value of FALSE.

Note: It is imperative to de-provision all accounts associated with a deleted user, because if you create a new user with the same user name as that of the deleted user by setting the `XL.UserIDReuse` property to TRUE, then the new user might get access to offline accounts of the deleted user that was not deleted as part of the de-provisioning process.

---

4. In addition to creating a non-unique index, create a unique functional index similar to the following:

```
create unique index UDX_USR_LOGIN_UNQ ON USR (USR_LOGIN,
ACT_KEY,DECODE(USR_STATUS,'Active',USR_STATUS,TO_CHAR(USR_KEY)));
```

   This index prevents the existence of multiple active users with the same login name, while permitting the existence of multiple deleted users with that login name. Without this unique index, it is possible in race conditions to create two active users with the same login name, if they are both created at the same time.

5. Click **Save**.

20.2.7 Configuring Delayed Delete Interval

To configure the delayed delete interval:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. Search for the `Period to Delay User Delete` system property with keyword `XL.UserDeleteDelayPeriod`.

   This property is used to specify the time period before deleting a user.
3. If you set a value of this property and a user is deleted, then the user's state is changed to disabled and "automatically delete on date" is set to current date plus the delay period.
4. If you do not set a value of this property, then the user is automatically deleted at the expiration of the end date by the `Disable/Delete User After End Date` scheduled job.
5. Save the system property value.

20.3 Configuring the Access Catalog

This section describes the following access catalog configurations:

- Configuring Additional Information
- Configuring Search Results
- Configuring the Sort By Attributes
- Configuring Custom Search

20.3.1 Configuring Additional Information

To configure additional information displayed in the access catalog:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. Search one or more of the following system properties depending on the entity for which you want to display additional information, and specify values.
   - **Catalog Additional Application Details Task Flow:** A custom task flow is to be displayed when an application is selected from the catalog checkout page. The task flow page will display as a tab in the cart details section.

     Replace the default value with the path to your custom task flow.

- **Catalog Additional Entitlement Details Task Flow:** A custom task flow is to be displayed when an entitlement is selected from the catalog checkout page. The task flow page will display as a tab in the cart details section.

  Replace the default value with the path to your custom task flow.

- **Catalog Additional Role Details Task Flow:** A custom task flow is to be displayed when a role item is selected from the catalog checkout page. The task flow page will display as a tab in the cart details section.

  Replace the default value with the path to your custom task flow.

3. Save the system properties.

20.3.2 Configuring Search Results

If you want to use searchable UDF in TAGS, then you can set the value of this property to TRUE. Then, you can run the scheduled task in recalculate tags mode and searchable UDF values are part of the TAGS column. The same value can be used in keyword search.

To configure the display of search results in the access catalog:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. To change the number of rows displayed in the tables in the access catalog, search for the `Catalog Table Rows To Display Size` system property, and specify the number of rows as the value.
3. If want to use searchable UDF in TAGS, then search for the `Catalog Searchable UDF In Tags` system property, and set the value to `TRUE`.
4. To control the maximum number of applications that can be selected for entitlement search, search for the `Catalog Advanced Search Maximum Applications` system property, and specify the number of applications.
5. Save the system properties.

20.3.3 Configuring the Sort By Attributes

To configure the attributes that you can use to sort the catalog search results:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.
2. Search for the `Catalog Attributes for Sorting Search Results` system property.
3. Specify the attributes that you want to be displayed in the Sort By drop down in the catalog results as the value of this property in the following format:

   ```
   ENTITY_DISPLAY_NAME; ENTITY_TYPE
   ```

4. Save the system property.

20.3.4 Configuring Custom Search

To customize catalog search, for example add search fields and search operators:

1. In the left pane of Oracle Identity System Administration, under System Configuration, click **Configuration Properties**.

2. Search for the `Catalog Advanced Search Taskflow` system property.
3. Replace the value of this system property with the complete path to the custom taskflow that you created. See "Customizing Catalog Search" in *Developing and Customizing Applications for Oracle Identity Manager* for information about creating the custom taskflow.
4. Save the system property.

20.4 Configuring the Identity Provider

This section describes the following identity provider configurations:

- Configuring Attribute Reservation
- Configuring Common Name Generation
- Configuring LDAP Reservation
- Configuring Referential Integrity

20.4.1 Configuring Attribute Reservation

To configure attribute reservation in Oracle Identity Manager:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Configuration Properties**.
3. Search for the `User Attribute Reservation Enabled` system property with keyword `XL.IsUsrAttribReservEnabled`.

   The default value of this `TRUE`, which means that user attribute reservation is enabled by default.
4. To disable user attribute reservation, modify the value of this property to `FALSE`.
5. Click **Save**.

20.4.2 Configuring Common Name Generation

To configure attribute reservation in Oracle Identity Manager:

1. Login to Oracle Identity System Administration
2. In the left pane, under System Configuration, click **Configuration Properties**.
3. Search for the `CommonName generation plugin` system property with keyword `XL.DefaultCommonNamePolicyImpl`.

   This property determines the common name generation plugin to generate common name. The default value is `oracle.iam.ldapsync.impl.plugins.FirstNameLastNamePolicy`.
4. Modify the value of this property to specify a different common name generation plugin.
5. Click **Save**.

> **See Also:** "Common Name Generation" in *Developing and Customizing Applications for Oracle Identity Manager* for more information

20.4.3 Configuring LDAP Reservation

LDAP Reservation Plugin

To configure attribute reservation in Oracle Identity Manager:

1. Login to Oracle Identity System Administration
2. In the left pane, under System Configuration, click **Configuration Properties**.
3. Search for the LDAP Reservation Plugin system property with keyword XL.LDAPReservationPluginImpl.

   This property determines the LDAP reservation plugin implementation to be picked up for reservation of user attributes.
4. Modify the value of this property to specify a different LDAP reservation plugin implementation for reservation of user attributes.
5. Click **Save**.

20.4.4 Configuring Referential Integrity

To configure attribute reservation in Oracle Identity Manager:

1. Login to Oracle Identity System Administration
2. In the left pane, under System Configuration, click **Configuration Properties**.
3. Search for the `Indicates if referential integrity is enabled in target LDAP directory` system property with keyword `XL.IsReferentialIntegrityEnabledInLDAP`.

   The default value of this property is FALSE, which means that referential integrity in the target LDAP directory is disabled.
4. To enable referential integrity in target the LDAP directory, modify the value of this property to `TRUE`.
5. Click **Save**.

21

Moving From Test to Production

Configurations and customizations in Oracle Identity Manager can be migrated from one deployment to another deployment. For example, you might want to migrate the configurations and customizations from a test environment to a production environment. This is referred to as Test to Production (T2P).

T2P can be performed in the following ways:

- **Incremental T2P:** In this type of T2P, you use the Deployment Manager tool for exporting and importing Oracle Identity Manager configurations and customizations. This is used when target/production setup is already configured and you want to move certain specific artifacts/configuration incrementally into the target setup.
- **Full T2P:** Fusion Middleware Framework-based movement scripts are used for this type of T2P. These scripts are used to move all the properties of an environment to another environment without the environment-specific attributes, which can be reconfigured. The full T2P process is done when you want to create a new production/target setup out of a test/source setup. During this process, all transactional and instance-specific data, such as users, provisioned/reconciled accounts, request data, reconciliation data, and audit data, is not moved to production setup, and the rest of the configurations/data are moved and made available for use on target setup.

> **Note:** Movement scripts support only Oracle WebLogic Application Server, and full T2P of Oracle Identity Manager on other application servers is not supported.

This chapter describes T2P in the following sections:

21.1 Migrating Incrementally Using the Deployment Manager

The Deployment Manager is a tool for exporting and importing Oracle Identity Manager configurations. 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 or a higher version must be installed on any computer that is running the Oracle Identity System Administration.

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 and binary data from the source system, store the information in an XML file, and then import the information from the XML file to the target system. The binary data includes plug-ins, JARs, and custom resource bundles.

An object exported from one type of repository is imported to the same type of repository.

Note: In addition to the Deployment Manager, you can use the sandbox feature to migrate configurations and customizations from one deployment to another. See "Managing Sandboxes" in *Developing and Customizing Applications for Oracle Identity Manager* for information about working with sandboxes.

This section includes the following topics:

- Features of the Deployment Manager
- Exporting Deployments
- Importing Deployments
- Best Practices Related to Using the Deployment Manager
- Troubleshooting the Deployment Manager

Note: Importing and exporting deployments by using the Deployment Manager can only be performed by the System Administrator.

21.1.1 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 configuration artifacts:

- Access policies
- Admin roles
- Application instances
- Approval policies
- Attestation processes
- Catalog metadata
- Certification configurations
- Certification definitions
- Custom resource bundles
- E-mail definitions
- Error codes
- Event handlers
- Generic Technology Connectors (GTC)
- GTC providers
- Identity Audit configuration
- Identity Audit rules
- Identity Audit scan definitions
- IT resource definition
- IT resources
- JAR files
- Lookup definitions
- Notification templates
- Organization metadata
- Organizations
- Password policies
- Policies
- Plug-ins
- Prepopulation adapters
- Process definitions
- Process forms
- Provisioning workflows and process task adapters
- Request datasets
- Resource objects
- Risk configuration
- Role metadata
- Roles
- Scheduled jobs

- Scheduled tasks
- System properties
- User metadata

Note:

- On the source, the following artifacts that are being exported might contain references to specific users, roles, application instances, entitlements, or organizations:
  - Certification definitions
  - Policies
  - Identity Audit configurations
  - Identity Audit scan definitions

  These specific references are scrubbed while exporting the artifacts and then importing them on the target setup. On the target, the artifact must be opened and updated for selection of these entities on the target. The artifacts cannot be used unless they are updated and will result in errors if used without updating. Any artifact that is generic and do not contain specific references can be used as it is after importing. For example, remediator name for Identity Audit policy is scrubbed off while export, and must be reselected on the target environment.
- All rules other than Identity Audit Rules are exported and imported implicitly with their policy by using Policy export/import and cannot be exported/imported independently because their existence is with their policy only.

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.

21.1.2 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.

Note: Application instances are exported and imported without the datasets. The datasets are migrated as a part of UI customization.

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: 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 System Administration.
2. In the left pane, under System Management, 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, Google Chrome, and Apple Safari web browsers.
- Apple Safari web browser overrides the applet security settings to impose restrictions on unsafe behavior, which stops any file reads/writes by applets. Therefore, run the applet in unsafe mode.

---

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 21-2 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.

    Select **Add More** and click **OK** to go to Search Objects Page to add more objects for export.

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 3 through 10. 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 Objects** 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**.

21.1.3 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. The user must be a Deployment Manager Administrator to be able to see Deployment Manager menu items on the UI based on menu permissioning model.
- 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.

To import an XML file:

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

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Management, click **Import**. The Deployment Manager opens.

   If another import from any other session is in progress, then a dialog box is displayed stating that the Deployment Manager import utility is currently used by another user. Click **Get Lock** to start the import process.

   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, Google Chrome, and Apple Safari Web browsers.

   ---

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 3 through 9.
    - 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.

21.1.4 Best Practices Related to Using the Deployment Manager

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

- Do Not Export System Objects
- Exporting Related Groups of Objects
- Using Logical Naming Conventions for Versions of a Form
- Exporting Root to Preserve a Complete Organizational Hierarchy
- Providing Clear Export Descriptions
- Checking All Warnings Before Importing
- Checking Dependencies Before Exporting Data
- Matching Scheduled Task Parameters
- Deployment Manager Actions on Reimported Scheduled Tasks
- Compiling Adapters and Enable Scheduled Tasks

- Checking Permissions for Roles
- Creating a Backup of the Database
- Importing Data When the System Is Quiet
- Exporting and Importing Data in Bulk
- Exporting Entity Publications

21.1.4.1 Do Not Export System Objects

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.

21.1.4.2 Exporting 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.

21.1.4.3 Using 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.

21.1.4.4 Exporting 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.

21.1.4.5 Providing 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.

21.1.4.6 Checking All Warnings Before Importing

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

21.1.4.7 Checking 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.

21.1.4.8 Matching Scheduled Task Parameters

Scheduled tasks depend on certain parameters to run properly. You can import scheduled task parameters to the production server. Table 21–1 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 21–1 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.

21.1.4.9 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 re-import:

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

21.1.4.10 Compiling 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.

21.1.4.11 Checking 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.

21.1.4.12 Creating a Backup of 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.

21.1.4.13 Importing 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.

21.1.4.14 Exporting and Importing Data in Bulk

The Deployment Manager is not a tool for data movement or migration of large volumes of data. Use your judgement while using the Deployment Manager to export/import objects. Entities, such as users, organizations, and roles, must be exported/imported by using other bulk tools, especially when the data volume is large.

In addition, ensure that users, roles, and organizations are always loaded and/or synchronized before moving of configuration objects, such as policies, rules, application instances, and connector configuration, to avoid exporting/importing them as dependencies.

Note: When exporting/importing large volumes of data, timeouts can occur in the UI.

21.1.4.15 Exporting Entity Publications

When exporting/importing an entity by using the Deployment Manager, any publication previously associated to the entity is removed, and no publication is assigned by default if you do not export the publication. For example, when you import an admin role that is published to an organization in the source environment, the admin role's publication information is lost in the target environment. Therefore, you must import the entity publication along with the admin role.

21.1.5 Troubleshooting the Deployment Manager

This section contains the following topics:

- Troubleshooting Deployment Manager Issues
- Enabling Logging for the Deployment Manager

21.1.5.1 Troubleshooting Deployment Manager Issues

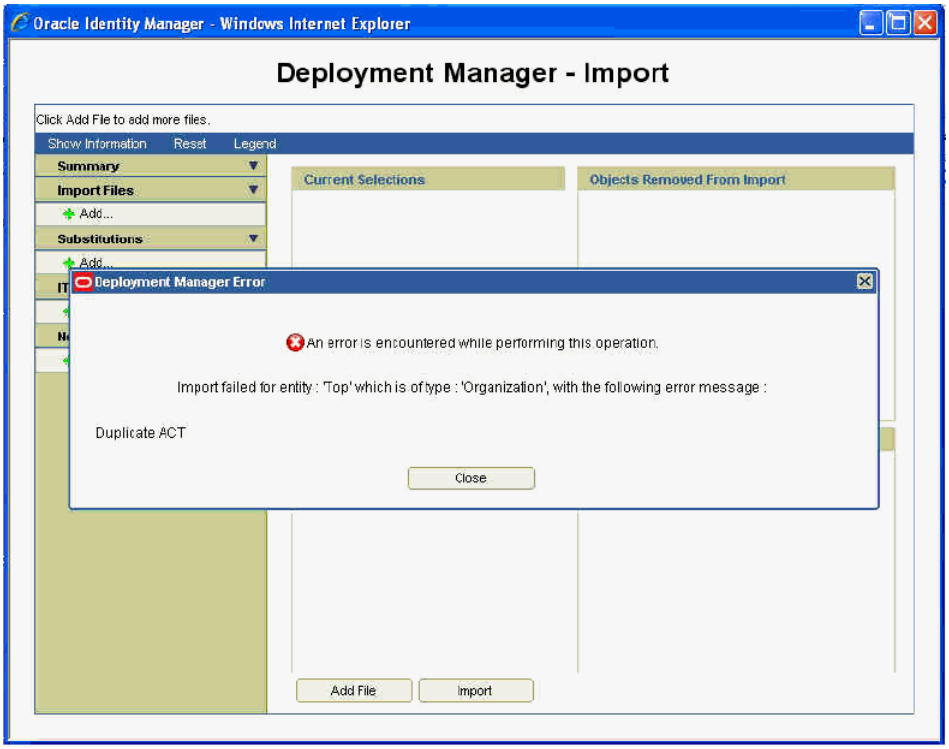
While importing data by using the Deployment Manager, the following information is displayed on the UI for an import failure:

- The entity for which the import failed
- The type of the entity for which the import failed
- The specific error message from the exception object

This information is also printed in logs along with the exception trace.

Figure 21–1 shows a sample error message that is displayed when Deployment Manager import fails.

Figure 21–1 Deployment Manager Import Failure



This helps the user in identifying which entity is causing the failure and why, and the user can try removing that particular entity and importing again if it is not necessary to be imported on the target system. This also helps the support team and developers in identifying the issue if it happens.

Table 21–2 lists the troubleshooting steps that you can perform if you encounter a failure:

Table 21–2 Troubleshooting Deployment Manager

Problem	Solution
In Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0), scheduled job has a dependency on scheduled task. Therefore, scheduled task must be imported prior to scheduled job. As a result, if a XML file has scheduled job entries prior to scheduled task entries, then importing the XML file using Deployment Manager fails with the following error message: `[exec] Caused By: oracle.iam.scheduler.exception.SchedulerExceptio n: Invalid ScheduleTask definition [exec] com.thortech.xl.ddm.exception.DDMException`	Open the XML file and move all scheduled task entries above the scheduled job entries.
Deployment Manager export fails for any object. User is prompted with Export Failed dialog box, and no exception is found in the server log. When you look at the JRE console, you can see the following: java.security.AccessControlException: access denied (java.io.FilePermission *PATH_AND_NAME_OF_THE_FILE*)	Perform the following steps: 1. Modify your java.policy in the *JRE_HOME*/lib/security/ directory. 2. Replace the existing policy file content with the following: `grant{ permission java.security.AllPermission; };` 3. Restart the browser to load the policy again. You can now export the data.
The following error occurs while importing an XML file: `Caused by: oracle.iam.reconciliation.exception.ConfigExcept ion: Profile :Xellerate User InvalidAttributes :`	Perform any one of the following: ■ Remove the attribute on which the error is generated from the XML, and then try importing. ■ Create the missing UDF or other attributes by using configuration service, and then retry the import. ■ Export the UDF shown as missing dependency. Import this UDF first before importing the current XML.
Importing approval policy might result in the following error: `weblogic.kernel.Default (self-tuning)'] [userId: xelsysadm] [ecid: f9e72ab2a292a346:-188377b2:12f96ae9676:-8000-000 0000000000047,0] [APP: oim#11.1.1.3.0] Exception thrown {0}[[ oracle.iam.platform.entitymgr.ProviderException: USER_NOT_FOUND`	An approval policy rule is invalid if it points to an entity (user or organization) that does not exist in Oracle Identity Manager. These invalid approval rules must be corrected to point to a valid entity (user or organization) before the import.

21.1.5.2 Enabling Logging for the Deployment Manager

To enable logging for the Deployment Manager:

1. Add a new logger for the Deployment Manager by editing the logging.xml file, which is located in the following directory path:

   DOMAIN_NAME/config/fmwconfig/servers/*SERVER_NAME*/

For instance, to enable Notification-level logging for Deployment Manager, add the following logger inside the <loggers> section:

```
<logger name='XELLERATE.DDM' level='NOTIFICATION:1' />
```

2. Change the log level defined in the relevent <log_handler>.

See Also: "Configuring Logging" on page 27-5 for information about logging level and log handlers in Oracle Identity Manager

21.2 Moving from a Test to a New Production Environment Using Movement Scripts

Oracle Identity Manager is a part of the Fusion Middleware environment. To move Oracle Identity Manager from test to production, you use the movement scripts. These scripts copy the Oracle Identity Manager binaries, artifacts, and configurations, and configures production Oracle Identity Manager with new end-points. The movement scripts interact with Oracle Identity Manager artifacts at the test and production environments and updates the production environment to make Oracle Identity Manager functional on the production environment. For detailed information about using the movement scripts, see "Moving from a Test to a Production Environment" in the *Oracle Fusion Middleware Administrator's Guide*. For the complete procedure for moving Oracle Identity Manager components, see "Moving Identity Management Components to a New Target Environment" in the *Oracle Fusion Middleware Administrator's Guide*.

Note: Before proceeding with migrating a source Oracle Identity Manager setup to a target setup, you can refer to "Limitations in Moving from Test to Production" in the *Oracle Fusion Middleware Release Notes* for information about the limitations and known issues related to moving from test to production. In addition, see "Troubleshooting Movement From Test to Production Environment Using Movement Scripts" on page 21-19 for information about the issues that you might encounter while migrating a source Oracle Identity Manager setup and the possible solutions.

For information about troubleshooting T2P issues applicable to an upgraded environment, see rows 6, 7, and 8 in Table 21–3, " Troubleshooting Movement From Test to Production Environment Using Movement Scripts".

To migrate a source Oracle Identity Manager setup to a target setup:

1. Migrate Oracle Identity Manager database schema data and embedded BI Publisher data from source to target DB host, as described in "Task 4 Perform Prerequisite Task for Oracle Identity Manager" under section "Moving Identity Management to a New Target Environment" in the *Oracle Fusion Middleware Administrator's Guide*.

Note: Data movement for components, such as SOA , MDS , and OPSS, is automated as part of the whole process described in this section, and no separate steps are required be done for them unless otherwise stated by the respective component.

---

2. Create the target setup by using the FMW T2P utilities. To do so:

   a. Run the following commands from the *ORACLE_COMMON_HOME*/bin/ directory.

Note:

- On Microsoft Windows, run the commands with .cmd extension, such as copyBinary.cmd and pasteBinary.cmd. For example, the copyBinary script is *ORACLE_COMMON_HOME*/bin/copyBinary.sh for UNIX and *ORACLE_COMMON_HOME*/bin/copyBinary.cmd for Microsoft Windows.
- Some arguments might be invalid for Windows operating system. For example, the -ipl *PATH_TO_ORACLE_INVENTORY_POINTER* argument does not work in Windows.
- This document provides the syntax for running the copyBinary, copyConfig, extractMovePlan, and pasteBinary scripts. For detailed information about these scripts, parameters, and example usages, see "Using the Movement Scripts" in the *Oracle Fusion Middleware Administrator's Guide*.

```
./copyBinary.sh -javaHome PATH_TO_JDK -al ARCHIVE_LOCATION -smw
SOURCE_MW_HOME -idw true -ipl PATH_TO_ORACLE_INVENTORY_POINTER -silent
true -ldl PATH_TO_LOG_DIRECTORY

./copyConfig.sh -javaHome PATH_TO_JDK -archiveLoc ARCHIVE_LOCATION
-sourceDomainLoc SOURCE_DOMAIN_LOCATION -sourceMWHomeLoc
MIDDLEWARE_HOME_LOCATION -domainHostName DOMAIN_HOST_NAME -domainPortNum
DOMAIN_PORT_NUMBER -domainAdminUserName DOMAIN_ADMIN_USERNAME
-domainAdminPasswordFile DOMAIN_ADMIN_PASSWORD_FILE -silent true -ldl
PATH_TO_LOG_DIRECTORY

./extractMovePlan.sh -javaHome PATH_TO_JDK -archiveLoc ARCHIVE_LOCATION
-planDirLoc MOVE_PLAN_DIRECTORY
```

In between running the extractMovePlan and pasteConfig scripts, update the moveplan with the new values for configuring the target. See "Modifying Move Plans" in the *Oracle Fusion Middleware Administrator's Guide* for information about common moveplan modifications. See the moveplan property descriptions in "Table 20-22 Move Plan Properties for Oracle Identity Manager" in the *Oracle Fusion Middleware Administrator's Guide*.

Note:

- While editing the moveplan, provide the listen address of the target in the Oracle Identity Manager Managed Server details.
- The datasource JDBC URL coming from source to the moveplan can either be in SID format, which is "jdbc:oracle:thin:@*HOST*:*PORT*:*SID*", or in service name format, which is "jdbc:oracle:thin:*HOST*:*PORT*/*SERVICE_NAME*". But you must always provide the JDBC URL in the datasource details in the service name format.

---

b. On the target host, create a new directory and copy pasteBinary.sh from the *SOURCE_MACHINE*/Middleware/oracle_common/bin/ directory. In addition, copy the cloningclient.jar file from the *SOURCE_MACHINE*/Middleware/oracle_common/jlib/ directory to the target host. Make sure that these two files are in the same location, for example /scratch/aime1/scripts. Then, run the following command:

```
./pasteBinary.sh -javaHome PATH_TO_JDK -al ARCHIVE_LOCATION -tmw
TARGET_MW_HOME -idw true -esp true -ipl PATH_TO_ORACLE_INVENTORY_POINTER
-ldl PATH_TO_LOG_DIRECTORY -silent true
```

c. Go to the *TARGET_MIDDLEWARE_HOME*/bin/ directory, and run the following command:

```
./pasteConfig.sh -javaHome PATH_TO_JDK -archiveLoc ARCHIVE_LOCATION
-targetDomainLoc TARGET_DOMAIN_PATH -targetMWHomeLoc
TARGET_MIDDLEWARE_HOME_PATH -movePlanLoc MOVE_PLAN_PATH
-domainAdminPasswordFile DOMAIN_ADMIN_PASSWORD_FILE -silent true -ldl
PATH_TO_LOG_DIRECTORY
```

Note:

- You might need to change the permissions on the *TARGET_MIDDLEWARE_HOME* and the target directory on which the JAR has been placed.
- Provide consistent directory paths for each of the parameters. For example, if you are using absolute path for *MIDDLEWARE_HOME*, then specify this path in the same way at all places.

3. Verify or modify the following configurations after full T2P migration:

- In the xlclient.cmd file, update the JDK path if the JDK library that was configured with the Design Console on the source is no longer accessible on the target.

  In the config/xlconfig.xml file, update the Application JNDI URL to point to the target application URL instead of source application URL.
- The IT Resource configurations are not part of the moveplan in the T2P procedure. After completing the T2P steps and starting the servers on the target setup, you can configure the IT Resource parameters as per the production setup. In Oracle Identity System Administration, under Configuration, click IT Resource. On the Manage IT Resource page, click the edit icon for the IT resource that you want to modify.
- Some entities, such as users and provisioned accounts, are not migrated from source to target during the T2P procedure, as they are considered transactional data. Therefore, user personalization settings such as sort order, saved searches, and layout changes will not be found on the target setup.
- Some users, such as role owners, are referenced in many places in Oracle Identity Manager. After full T2P migration, references to such users are replaced with reference to *SYSTEM_ADMINISTRATOR_USERNAME*, the Oracle Identity Manager system administrator.
- Check the following functionality on the target setup after the T2P process is complete:

  - User creation, by assigning an existing role tied to an access policy (role and access policy migrated from source)
  - Role creation, both enterprise role and admin role
  - Request creation and approval workflow initiation
  - Basic OIM-OAM and OIM-OAAM integration usecases if topology has the integration defined
- Customization done on the LDAP sync configuration, such as role category container rules has to be configured on the T2P environment post migration.

21.2.1 Troubleshooting Movement From Test to Production Environment Using Movement Scripts

Table 21–3 lists the troubleshooting step that you can perform if you encounter issues related to movement from a test to a new production environment by using movement scripts.

Table 21–3 Troubleshooting Movement From Test to Production Environment Using Movement Scripts

#	Problem	Solution
1	After migrating from an Oracle Identity Manager clustered deployment to another, SOA Server is running but you are not able to access soa-infra. This is because the coherence settings are pointing to source.	Change the coherence settings accordingly by referring to "Specifying the Host Name Used by Oracle Coherence" in the *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management*, and then restart the SOA Server.

2	The following section is logged in the cloning error logs:
	`NOTIFICATION: PManager instance is created without multitenancy support as JVM flag "oracle.multitenant.enabled" is not set to enable multitenancy support.`
	`Sep 24, 2013 10:26:55 PM oracle.security.jps.internal.config.xml.XmlConfigurationFactory initDefaultConfiguration`
	`SEVERE: java.io.FileNotFoundException: ./config/jps-config.xml (No such file or directory)`
	`Sep 24, 2013 10:26:55 PM oracle.mds NOTIFICATION: Auditing is disabled for component MDS.`
	`Sep 24, 2013 10:26:55 PM oracle.mds NOTIFICATION: PManager instance is created without multitenancy support as JVM flag "oracle.multitenant.enabled" is not set to enable multitenancy support.`
	`Sep 24, 2013 10:26:55 PM oracle.security.jps.internal.config.xml.XmlConfigurationFactory initDefaultConfiguration`
	`SEVERE: java.io.FileNotFoundException: ./config/jps-config.xml (No such file or directory)`
	`Sep 24, 2013 10:26:55 PM oracle.mds NOTIFICATION: Auditing is disabled for component MDS.`
	`Sep 24, 2013 10:26:55 PM oracle.mds`
		This section of the cloning error logs is benign and can be safely ignored.

Table 21–3 (Cont.) Troubleshooting Movement From Test to Production Environment Using Movement

#	Problem	Solution

3	After migrating Oracle Identity Manager to a new environment, database connection-related errors are thrown when you try one or more of the following operations:
	■ Create a user
	■ Search and open a user
	■ Provision an application instance to a user
		Set the following tuning parameters as shown:
		`JAVA_OPTIONS="-Djbo.ampool.doampooling=true -Djbo.ampool.minavailablesize=1`
		
		`-Djbo.ampool.maxavailablesize=120`
		`-Djbo.recyclethreshold=60`
		`-Djbo.ampool.timetolive=-1`
		`-Djbo.load.components.lazily=true`
		`-Djbo.doconnectionpooling=true`
		`-Djbo.txn.disconnect_level=1`
		`-Djbo.connectfailover=false`
		`-Djbo.max.cursors=5`
		`-Doracle.jdbc.implicitStatementCacheSize=5`
		`-Doracle.jdbc.maxCachedBufferSize=19`
		`${JAVA_OPTIONS}"`
		
		For information about these tuning parameters, see "Application Module Pooling" in the *Oracle Fusion Middleware Performance and Tuning Guide*.

4	The following type of error is thrown:
	`Caused By:`
	`java.sql.SQLIntegrityConstraintViolationException:`
	`ORA-00001:`
	`unique constraint (SOURCE_OIM.PK_ORD) violated`
		Check impdp logs to see if there area any errors there that are not listed as `to be ignored`.

5	The following error is thrown:
	`UserConfigDataMigrationException:`
	`oracle.bpel.services.workflow.util.tools.wfUserConfigDataMigrator.UserConfigDataMigrationException: ORABPEL-30511`
	`Verification Service cannot resolve user identity.`
	`User weblogic cannot be found in the identity repository.`
	`Workflow Context token cannot be null in request.`
		Check if you have used the required parameters and parameter file during Oracle Identity Manager schema migration.

Table 21–3 (Cont.) Troubleshooting Movement From Test to Production Environment Using Movement

#	Problem	Solution

6	After migrating to the target environment, the following errors are encountered during the create user or role operation:
	
	Error in UI:
	
	`An Error Occured while deleting LDAP User in the compensate stage`
	
	Error in the logs:
	
	`An error occurred while removing the entity in LDAP, and the corresponding error is - {0}[[oracle.iam.platform.entitymgr.vo.ConnectivityExcepti on: java.lang.IllegalArgumentException: Null input buffer at oracle.iam.ldapsync.impl.repository.ITResourceReposi tory.getConnection(ITResou rceRepository.java:40) at oracle.iam.platform.entitymgr.provider.ldap.LDAPData Provider.remove(LDAPDataPr ovider.java:1170) at oracle.iam.platform.entitymgr.impl.EntityManagerImpl .deleteEntity(EntityManage rImpl.java:704) at oracle.iam.platform.entitymgr.impl.EntityManagerImpl .deleteEntity(EntityManage rImpl.java:675) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMe thodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(Dele gatingMethodAccessorImpl.j ava:25) at java.lang.reflect.Method.invoke(Method.java:597) at org.springframework.aop.support.AopUtils.invokeJoinp ointUsingReflection(AopUti ls.java:307) at org.springframework.aop.framework.ReflectiveMethodIn vocation.invokeJoinpoint(R eflectiveMethodInvocation.java:182) at org.springframework.aop.framework.ReflectiveMethodIn vocation.proceed(Reflectiv eMethodInvocation.java:149) at oracle.iam.platform.utils.DMSMethodInterceptor.invok e(DMSMethodInterceptor.jav a:35) ... ... Caused by: java.lang.IllegalArgumentException: Null input buffer``
		Perform the following steps:
		
		1. Connect to OIM DB schema and run the following SQL query:
		
		`UPDATE SVP SET SVP_FIELD_VALUE=NULL WHERE SVR_KEY IN (Select SVR_KEY from SVR WHERE SVR_NAME = 'Directory Server') AND SPD_KEY IN (SELECT SPD_KEY FROM SPD WHERE SVD_KEY IN (Select SVD_KEY from SVR WHERE SVR_NAME = 'Directory Server') AND SPD_FIELD_NAME IN('Admin Login','Admin Password','User Reservation Container','Search Base','Server URL','Use SSL'));`
		
		2. Login to Oracle Enterprise Manager.
		
		3. Go to **System MBean Browser**, **Application Defined MBeans**, **oracle.iam**, **IAMAppRuntimeMBean**, **IDStoreConfigMBean**.
		
		4. Set the values for the following parameters:
		
		Admin Login
		
		Admin Password
		
		User Reservation Container
		
		Search Base
		
		Server URL
		
		Use SSL

Table 21–3 (Cont.) Troubleshooting Movement From Test to Production Environment Using Movement

#	Problem	Solution
7	After migrating to the target environment, the following error is encountered while performing the role update operation: Error in UI: `IAM-3056030 : An exception occurred while performing the operation.` Error in the logs: `[2015-04-01T22:57:17.451-07:00] [oim_server1] [WARNING] [] [oracle.adf.controller.faces.lifecycle.Utils] [tid: [ACTIVE].ExecuteThread: '3' for queue: 'weblogic.kernel.Default (self-tuning)'] [userId: xelsysadm] [ecid: 0000Klsh7XMFW7KpISP5if1L6xnV0000^G,0] [APP: oracle.iam.console.identity.self-service.ear#V2.0] [DSID: 0000Klsbh6yFW7KpISP5if1L6xnV0000Yj] ADF: Adding the following JSF error message: IAM-3056030 : An exception occurred while performing the operation.[[oracle.iam.ui.platform.exception.OIMRuntimeException : IAM-3056030 : An exception occurred while performing the operation.` `at oracle.iam.ui.platform.exception.OIMErrorHandler.reportServiceException(OIMErrorHandler.java:178)` `at oracle.iam.ui.platform.exception.OIMErrorHandler.reportException(OIMErrorHandler.java:66)` `at oracle.adf.model.binding.DCDataControl.reportException(DCDataControl.java:413)` `.` `at oracle.adf.model.binding.DCBindingContainer.reportException(DCBindingContainer.java:425)` `at oracle.adf.model.binding.DCBindingContainer.reportException(DCBindingContainer.java:480)` `at oracle.adf.model.binding.DCControlBinding.reportException(DCControlBinding.java:201)` `at oracle.jbo.uicli.binding.JUCtrlActionBinding.reportException(JUCtrlActionBinding.java:2101)` `at oracle.jbo.uicli.binding.JUCtrlActionBinding.doIt(JUCtrlActionBinding.java:1733)` `at oracle.adf.model.binding.DCDataControl.invokeOperation(DCDataControl.java:2188)` `at`	Connect to Oracle Identity Manager database schema, and run the following SQL query: `update ugp set ugp_role_owner_key = (select usr_key from usr where usr_login = 'XELSYSADM' ) where ugp_role_owner_key is null;`

Table 21–3 (Cont.) Troubleshooting Movement From Test to Production Environment Using Movement

#	Problem	Solution

8	After migrating to the target environment, request creation fails, and the following error is encountered:
	
	Error in UI:
	`Request not raised, no other error seen as such`
	
	Error in the logs:
	`<Apr 1, 2015 5:28:17 AM PDT> <Error> <oracle.iam> <BEA-000000> < ORABPEL-30509 . Insufficient privileges to authenticate on behalf of another user. User weblogic cannot authenticate on behalf of user xelsysadm without admin privileges. Only users with admin privileges can authenticate on behalf of another user. . at weblogic.rjvm.ResponseImpl.unmarshalReturn(ResponseImpl.java:237) at weblogic.rmi.cluster.ClusterableRemoteRef.invoke(ClusterableRemoteRef.java:348) at weblogic.rmi.cluster.ClusterableRemoteRef.invoke(ClusterableRemoteRef.java:259) at oracle.bpel.services.workflow.query.ejb.TaskQueryService_oz1ipg_EOImpl_1036_WL Stub.authenticateOnBehalfOf(Unknown Source) at oracle.bpel.services.workflow.query.client.TaskQueryServiceRemoteClient.authen ticateOnBehalfOf(TaskQueryServiceRemoteClient.java:63) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.j ava:25) at java.lang.reflect.Method.invoke(Method.java:597) at oracle.bpel.services.workflow.client.WFClientRetryInvocationHandler.invokeTarg et(WFClientRetryInvocationHandler.java:144) at oracle.bpel.services.workflow.client.WFClientRetryInvocationHandler.invoke(WFC lientRetryInvocationHandler.java:82) at $Proxy436.authenticateOnBehalfOf(Unknown Source)`
		Connect to Oracle Identity Manager database schema, and run the following SQL query:
		
		`update usg set ugp_key = (select ugp_key from ugp where ugp_name = 'Administrators' ) where usr_key = ( select usr_key from usr where usr_login = 'WEBLOGIC') and ugp_key != (select ugp_key from ugp where ugp_name = 'ALL USERS') and ugp_key not in (select ugp_key from ugp);`

Part VIII

Auditing and Reporting

This part describes auditing and reporting features of Oracle Identity Manager.

It contains the following chapters:

22

Configuring Auditing

Oracle Identity Manager provides a powerful audit engine to collect extensive data for audit and compliance purposes. You can use the audit functionality together to capture, archive, and view entity and transactional data for compliance monitoring and IT-centric processes and forensic auditing.

This chapter consists of the following topics:

22.1 Overview

Oracle Identity Manager provides auditing and historical archiving of profile information. It takes a snapshot of a profile, stores the snapshot in an audit table in the database, and updates the snapshot each time the profile data changes. In the context of profile auditing, the term snapshot means a copy taken of the entire profile data at any instant when the data is modified.

22.2 User Profile Auditing

User profile audits cover changes to user profile attributes, user membership, resource provisioning, access policies, and resource forms.

This section discusses the following topics:

22.2.1 Data Collected for Audits

By default, user profile auditing is enabled and the auditing level is set to Resource Form when you install Oracle Identity Manager. This auditing level specifies the minimum level required for attestation of form data.

You configure the audit level in the System Configuration part of the Advanced Administration by using the XL.UserProfileAuditDataCollection system property.

> **See Also:** "System Properties in Oracle Identity Manager" on page 20-1 for information about the XL.UserProfileAuditDataCollection system property

The supported audit 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.

This section discusses the following topics:

22.2.1.1 Capture of User Profile Audit Data

Each time a user profile changes, Oracle Identity Manager takes a snapshot of the user profile and stores the snapshot in an audit table in the database.

A snapshot is also generated when there is a change in a user profile that must be audited, even if an initial snapshot is missing. The current snapshot is treated as the initial snapshot.

The following are the components of a user profile and the tables that store these components:

- **User Record:** Contains the USR table, including all User Defined Fields (UDFs).

  The USR table stores user attributes. When you create a user, Oracle Identity Manager adds an entry to this table.

- **User Role Membership:** Contains the UGP table.

  The UGP_USER_MEMBERSHIP_RULE column of the UGP table stores the user role membership rule.

  Role Manager API method *SearchRule getUserMembershipRule(String roleKey)* can be used to retrieve the user membership rule from UGP table.

  For more information about Role Manager APIs, see *Oracle Fusion Middleware Java API Reference for Oracle Identity Manager*.

- **Admin Role Membership:** Contains the ARM_AUD table.

The ARM_AUD table defines the membership ID, admin role ID, user ID, organization, hierarchy, action, and logged-in user.

- **User Policy Profile:** Contains the following tables:
  - **UPD:** Stores User Policy Profile data. This is a policy-centric view of the resources that are provisioned to a user.
  - **UPP:** Stores User Policy Profile-related details. This is a user-centric view of all the applicable policies for a user, and the resources they allow/deny.

> **Note:** When you change a role name by using Oracle Identity Self Service, the User Profile Audit (UPA) tables in the database are not updated with the change until the next snapshot of the user.

- **User Resource Profile:** This component can be divided into the following subcomponents:
  - **User Resource Instance:** Contains the OBI, OBJ, and OIU tables, as listed in Table 22–1.

Table 22–1 User Resource Instance Tables

Table Name	Description
OBI	Stores resource (object) instance information. Oracle Identity Manager creates a resource instance every time a resource is provisioned. This instance stores all generic information related to that provisioned instance, including a request key (if the resource has been provisioned through a request), the corresponding process instance, and the instance status.
OBJ	Represents the resource object data, including details about the resource, such as resource name, whether or not auto-save and auto-prepopulate are set, and whether or not the resource object allows multiple instances.
OIU	Associates applicable user information to the resource object instance when provisioning takes place. In addition, it stores policy-related information for the specific resource instance.

  - **Resource Lifecycle (Provisioning) Process:** Contains the MIL, ORC, OSI, PKG, SCH, and TOS tables, as listed in Table 22–2.

Table 22–2 Resource Lifecycle Process Tables

Table Name	Description
MIL	Defines the process task definitions. Each entry corresponds to a process task. A process definition (PKG table) comprises of multiple tasks, which are a part of the various workflows in the definition.
ORC	Stores process instance information when provisioning takes place. When provisioning starts, Oracle Identity Manager generates an associated process (or workflow) instance that stores process-related information specific to the provisioning instance.
OSI	Stores information about tasks created for process instance.

Table 22–2 (Cont.) Resource Lifecycle Process Tables

Table Name	Description
PKG	Defines processes or workflows in Oracle Identity Manager, including process details such as process name, process type, descriptive field mapping, and associated resources and process forms.
SCH	Stores information related to running of a specific task instance such as the task status, status bucket, and timing of when the adapter run started or ended.
TOS	Stores atomic process information.

– **Resource State (Process) Form:** This information is stored in the UD parent and child tables. The UD_* tables are user-defined field tables that store the account state.

22.2.1.2 Storage of Snapshots

When Oracle Identity Manager takes a snapshot of a user profile, it stores the snapshot in the UPA table. The structure of the UPA table is described in Table 22–3.

Table 22–3 Definition of the UPA Table

Column	Data Type	Description
UPA_KEY	NUMBER (19,0)	Key for the audit record
USR_KEY	NUMBER (19,0)	Key for the user whose snapshot is recorded in this entry
EFF_FROM_DATE	TIMESTAMP (6)	Date and time at which the snapshot entry became effective
EFF_TO_DATE	TIMESTAMP (6)	Date and time at which the snapshot entry was no longer effective In other words, this is the date and time at which the next snapshot entry was created. For the entry representing the latest user profile, the To Date column value is set to NULL.
SRC	VARCHAR2 (4000)	User ID of the user responsible for the change, and the API used to carry out the change
SNAPSHOT	CLOB	XML representation of the snapshot
DELTAS	CLOB	XML representation of old and new values corresponding to a change made to the snapshot
SIGNATURE	CLOB	Can be used to store a digital signature for the snapshot (for nonrepudiation purposes)

Note: The initial audit snapshots for default users in Oracle Identity Manager is not UTF-8 encoded. However, auditing of subsequent modifications to these users have UTF-8 encoded snapshots.

22.2.1.3 Trigger for Taking Snapshots

When any data element in a user profile changes, Oracle Identity Manager creates a snapshot.

The following events trigger the creation of a user profile snapshot:

- Modification of any kind to the user record (for example, through reconciliation and direct provisioning)

- Role membership change for the user
- Changes in the policies that apply to the user
- Provisioning a resource to the user
- Deprovisioning of a resource for the user
- Any provisioning-related event for a provisioned resource:
  - Resource status change
  - Addition of provisioning tasks to the provisioning process
  - Updates to provisioning tasks in the provisioning process, for example, status changes, escalations, and so on
  - Creation of or updates to Process Form data

22.2.2 Post-Processor Used for User Profile Auditing

The user profile auditor has an internal post-processor that normalizes the snapshot XML into the reporting tables: UPA_USR, UPA_FIELDS, UPA_GRP_MEMBERSHIP, UPA_RESOURCE, UPA_UD_FORMS, and UPA_UD_FORMFIELDS. These tables are used by the reporting module to generate the appropriate reports.

22.2.3 Tables Used for User Profile Auditing

Table 22–4 lists the tables in the database that User profile audits use:

> **Note:** For more information about the User Profile Audits tables, such as column names and how to use them, refer to the schema documentation provided with Oracle Identity Manager.

Table 22–4 User Profile Audit Tables

Table Name	Description
AUD	Stores detailed information about all of the Auditors (for example, the User Profile Auditor) supported by Oracle Identity Manager.
AUD_JMS	Staging table that stores information about changes made as a part of any business transaction. This is an intermediate table to temporarily store data changelog data before the audit engine consumes it. When Audit messages are successfully processed, corresponding records are deleted from the table. **Note:** This table is not intended for end users and must not be used directly.
UPA	Main auditing table for storing all snapshots and changes made to the user profiles.
UPA_FIELDS	Stores user profile audit history changes in denormalized (vertical) format.
UPA_GRP_MEMBERSHIP	Stores groups membership history in denormalized format.
UPA_RESOURCE	Stores user profile resource history in denormalized format.
UPA_USR	Stores user profile history in denormalized format.

Table 22–4 (Cont.) User Profile Audit Tables

Table Name	Description
UPA_UD_FORMS	Together with the UPA_UD_FORMFIELDS table, contains information about changes to the user's account profile (process form). This table keeps track of the changes to the various forms, such as parent or child forms, which are being changed in any transaction. The changes to the account or entitlement attributes are stored in the UPA_UD_FORMFIELDS table.
UPA_UD_FORMFIELDS	Stores the names of account or entitlement profile fields that are modified. This table also keeps track of the old and new values of the modified fields.

Note:

- The UPA_UD_FORMS and UPA_UD_FORMFIELDS tables together store the audit trail of changes to the user's account profile in a de-normalized format. These tables can be used in various audit-related reports.
- The UPA_UD_FORMS and UPA_UD_FORMFIELDS tables are populated only if the XL.EnableExceptionReports system property is set to TRUE. For more information about this property, see "System Properties in Oracle Identity Manager" in the *Oracle Fusion Middleware Administering Oracle Identity Manager*.
- The Form Upgrade Job schedule task updates the form version to the latest active version and the form data to the value specified during the field's creation for all accounts. If this scheduled task is not run, then the form version and data is incorrect in the audit snapshot and the reporting tables.

22.2.4 Archival

User Profile audit data growth is based on the setting of the audit levels, and the growth can be significant in most of the deployments.

There is also a requirement to clean or archive the old user profile audit data to accommodate future growth.

You can use Audit Archival and Purge Utility to meet these requirements. See "Using the Audit Archival and Purge Utility" on page 24-24 for detailed information about this utility.

22.3 Role Profile Auditing

Role profile audits cover changes to role profile attributes, role administrators, and direct subroles.

This section discusses the following topic:

- Data Collected for Audits

22.3.1 Data Collected for Audits

Role profile auditing is determined by the value of the Level of Role Auditing system property with keyword XL.RoleAuditLevel. This property controls the amount of

audit data collected when an operation is performed on a role, such as creation or modification. The supported levels are:

- **None:** No audit data is collected. This is the default value.
- **Role:** Creation, modification, and deletion of role is audited.
- **Role Hierarchy:** Changes made to the role inheritance is audited.

This section discusses the following topics:

- Capture and Archiving of Role Profile Audit Data
- Storage of Snapshots
- Trigger for Taking Snapshots

22.3.1.1 Capture and Archiving of Role Profile Audit Data

Each time a role profile changes, Oracle Identity Manager takes a snapshot of the role profile and stores the snapshot in an audit table in the database.

Oracle Identity Manager generates a snapshot when an audit is created for a role, even if an initial snapshot is missing. The current snapshot is treated as the initial snapshot.

The following are the components of a role profile and the tables that constitute these components:

- UGP: Role record, including all UDFs for roles
- GPG: Subrole/parent role information

22.3.1.2 Storage of Snapshots

When Oracle Identity Manager takes a snapshot of a role profile, it stores the snapshot in a GPA table. The structure of this table is as described in Table 22–5.

Table 22–5 Definition of the GPA Table

Column	Data Type	Description
GPA_KEY	NUMBER (19,0)	Key for the audit record
UGP_KEY	NUMBER (19,0)	Key for the role whose role snapshot is recorded
EFF_FROM_DATE	TIMESTAMP (6)	Date and time at which the snapshot entry became effective
EFF_TO_DATE	TIMESTAMP (6)	Date and time at which the snapshot entry was no longer effective In other words, this is the date and time at which the next snapshot entry was created. For the entry representing the latest user profile, the To Date column value is set to NULL
SRC	VARCHAR2 (4000)	Source of the entry, User ID of the user responsible for the change, and the API used to carry out the change
SNAPSHOT	CLOB	XML representation of the snapshot
DELTAS	CLOB	XML representation of old and new values corresponding to a change made to the snapshot
SIGNATURE	CLOB	Can be used to store a digital signature for the snapshot (for nonrepudiation purposes)

When Oracle Identity Manager takes a snapshot of a admin role membership profile, it stores the snapshot in the ARM_AUD table. The structure of this table is as described

in Table 22–6.

Table 22–6 Definition of the ARM_AUD Table

Column	Data Type	Description
ARM_AUD_ID	NUMBER(20)	Admin role audit ID
MEMBERSHIP_ID	NUMBER(20)	Admin role membership ID
ROLE_ID	NUMBER(20)	Admin role ID
USER_ID	VARCHAR2(256 CHAR)	User member ID
SCOPE_ID	NUMBER(20)	Scope ID of the admin role membership
INCLUDE_HIERARCHY	NUMBER(1)	Whether or not organization hierarchy is included
ARM_AUD_EFF_FROM_DATE	TIMESTAMP(6)	Date from which admin role audit is effective
ARM_AUD_EFF_TO_DATE	TIMESTAMP(6)	Date up to which admin role audit is effective
USR_ACTION	VARCHAR2(10 CHAR)	Action of the user members
USR_LOGIN	VARCHAR2(256 CHAR)	Login names of the user members

22.3.1.3 Trigger for Taking Snapshots

When any data element in the role profile snapshot changes, Oracle Identity Manager creates a snapshot.

The creation of role profile snapshots is triggered by events that result in changes in any of the following:

- Role profile data
- Parent role information
- Adding or revoking users or user memberships

22.4 Catalog Auditing

See "Configuring Catalog Auditing" on page 13-17 for information about catalog auditing.

22.5 Enabling and Disabling Auditing

This section describes how to enable and disable auditing in Oracle Identity Manager in the following sections:

- Disabling Auditing
- Enabling Auditing

22.5.1 Disabling Auditing

To disable auditing in Oracle Identity Manager:

1. Set the values of User profile audit data collection level (XL.UserProfileAuditDataCollection) and Level of Role Auditing (XL.RoleAuditLevel) system properties to None, as described in "Modifying System Properties" on page 20-23.

2. Disable the Issue Audit Messages Task scheduled job as described in "Disabling and Enabling Jobs" on page 18-24.

   If pending audit changes are required to be recorded in the audit tables, then disable the scheduled task after all the pending audit changes are processed.

To disable catalog auditing, set the value of the XL.CatalogAuditDataCollection system property to `none`.

22.5.2 Enabling Auditing

To enable auditing in Oracle Identity Manager:

1. Set the values of User profile audit data collection level (XL.UserProfileAuditDataCollection) and Level of Role Auditing (XL.RoleAuditLevel) system properties to one of the levels defined in "User Profile Auditing" on page 22-1 and "Role Profile Auditing" on page 22-6 for user profile and role profile auditing respectively.

   See "Configuring Oracle Identity Manager" on page 20-1 for information about modifying the values of system properties.

2. Enable the Issue Audit Messages Task scheduled job as described in "Disabling and Enabling Jobs" on page 18-24.

3. For user profile auditing, generate snapshots by running the GenerateSnapshot script as described in "Generating an Audit Snapshot" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

   The following is the command-line usage of the GenerateSnapshot script:

   ```
   ./GenerateSnapshot.sh -username OIM_ADMIN_USERNAME -numOfThreads 8 -serverURL
   t3://WLS_SERVER:PORT -ctxFactory weblogic.jndi.WLInitialContextFactory
   [-inputFile fileWithUserKeys]
   ```

   Here:

   - *OIM_ADMIN_USERNAME* is the Oracle Identity Manager administrator username.
   - *WLS_SERVER* is the Oracle WebLogic Server name.
   - *PORT* is the port number of the WebLogic Server.

To enable catalog auditing, set the value of the XL.CatalogAuditDataCollection system property to `catalog`.

22.6 Lightweight Audit

Oracle Identity Manager introduces Lightweight Audit Engine in this release.

The Lightweight Audit engine is direct. Unlike legacy auditing, it does not have a denormalized reporting schema. It is completely independent of the legacy audit engine. For backward compatibility, Oracle Identity Manager uses both the audit engines.

Table 22–7 lists the entities supported by the lightweight audit engine and the corresponding entity operations.

Table 22–7 Entities Audited by Lightweight Audit Engine

Entity	Operation
User	Create, Modify, Delete, Enabled, Disabled, Lock, Unlock, Reset Password, Change Password
Role	Create, Modify, Delete
Role-User Membership	Add, Remove, Modify
Organization	Create, Modify, Delete
Organization-User Membership	Added, Removed
Policy	Create, Modify, Delete, Enabled, Disabled
Rule	Create, Modify, Delete, Enabled, Disabled
Return Value (Child)	Create, Modify, Delete
Rule-Return Value (Child)	Add, Remove
Policy-Rule Relationship	Add, Remove
Policy Violation	Create, Modify, Delete, Enabled, Disabled
Policy Violation Cause	Create, Modify, Delete, Enabled, Disabled
Scan Definition	Create, Modify, Delete, Enabled, Disabled
Scan Definition-Policy Relationship	Create, Modify, Delete, Enabled, Disabled
Scan Run	Create, Modify, Delete, Enabled, Disabled
Scan Run-Policy Relationship	Add, Remove
Scan Run-User Relationship	Add, Remove
Scan Run-Policy Violation Relationship	Add, Remove
Remediator	Create, Modify, Delete
Task Policy Violation	Create, Modify, Delete

When Oracle Identity Manager records the changes to an entity or relationship, it stores the data in the AUDIT_EVENT table. Table 22–8 lists the attributes of the AUDIT_EVENT table.

Table 22–8 Definition of the AUDIT_EVENT Table

Attribute	Data Type	Description
event_id	VARCHAR2(40)	Unique ID of the audit log event
event_action	VARCHAR2(255)	Set of entity type actions such as CREATE, MODIFY, DELETE, ADD, REMOVE, MODIFY_RULE, and DELETE_RULE.
event_date	TIMESTAMP	Date of event
event_actor_id	VARCHAR2(40)	ID of the user who performed the action
event_actor_name	VARCHAR2(255)	Name of the user who performed the action
event_mechanism	VARCHAR2(40)	Self, Admin, Recon, Policy-Based, Request
event_request_id	VARCHAR2(40)	If mechanism is Request, then requestId of request

Table 22–8 (Cont.) Definition of the AUDIT_EVENT Table

Attribute	Data Type	Description
event_status	VARCHAR2(1)	Status of request S (success) or F (failure)
event_fail_reason	VARCHAR2(255)	Descriptive reason for failed action. For example, POLICY_VIOLATION, ACCOUNT_LOCKED, and REQUEST NOT APPROVED.
event_hash	VARCHAR2(1024)	When an audit event is created, the AuditEventManager generates a hash value based on the contents of the event like, example entity type, action, date/time and so on. The hash is encrypted and stored in event_hash. The AuditEventManager provides an API method to verify that a specific audit event's content has not been altered since the event was stored in the database. The method implementation retrieves the hash value and decrypts it. Then, the hash value is recomputed from the event's content and compared to the stored value. If the values are the same the method returns true, otherwise false.
event_values_added	CLOB	Values added. Data is in name1=value1 name2=value2 format, where name can be a simple name (for example firstname, userMembers) or a path expression to represent complex, nested attributes (for example user.address[type=work].street) - enables SQL "contains" search
event_values_removed	CLOB	Values removed. Data is in name1=value1 name2=value2 format, where name can be a simple name (for example firstname, userMembers) or a path expression to represent complex, nested attributes (for example user.address[type=work].street) - enables SQL "contains" search
entity_type	VARCHAR2(40)	Type of entity
entity_id	VARCHAR2(40)	ID of entity
entity_name	VARCHAR2(255)	Name of the entity, for example User Login, Role Name, or Policy Name.
to_entity_type	VARCHAR2(40)	Type of entity in relationship (for example User in Role-User, Rule in Policy-Rule, and so on)
to_entity_id	VARCHAR2(40)	ID of to entity in relationship
to_entity_name	VARCHAR2(255)	Name of to entity in relationship

23

Using Reporting Features

This chapter includes the following sections:

23.1 Reporting Features

Oracle Business Intelligence (BI) Publisher is Oracle's primary reporting tool for authoring, managing, and delivering all your highly formatted documents. BI Publisher is shipped with Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0).

BI Publisher is deployed and configured as a separate managed server within the same Oracle Identity Manager domain.

You have the choice of either leveraging the embedded BI Publisher or a standalone BI Publisher. It is recommended that you use the embedded BI Publisher if there are no other reporting requirements and you only need reporting for Oracle Identity Manager.

After BI Publisher configuration, you can take advantage of the standard features of BI Publisher, such as:

- Highly formatted and professional quality reports with pagination and headers/footers.
- PDF, Word, and HTML output of reports.
- Capability to develop your own custom reports against the Oracle Identity Manager repository (read-only repository access).
- BI Publisher's scheduling capabilities and delivery mechanisms, such as e-mail and FTP.
- Format (report) can be edited separately from the data definition (data model).
- Standardized Oracle Identity subtemplate for headers.

- National Language Support (NLS) for BI Publisher report output.

23.2 Starting Oracle Identity Manager Reports

Start BI Publisher server explicitly as managed server from the Oracle WebLogic Server Administrative Console.

To start BI Publisher:

1. Login to the WebLogic Administrative Console.
2. Under the Domain Structure, expand **Environment**.
3. Click **Servers** to display the Summary of Servers table.
4. Click **Control**. Select the BI Publisher server, for example bi_server1, and then click **Start**.

To start BI Publisher from the command line, run the following command:

```
$ ./startManagedWebLogic.sh BI_SERVERNAME t3://HOST_NAME:PORT
```

23.3 Supported Output Formats

BI Publisher supports multiple report output formats. All reports are generated in a native XML format which can be transformed into different other output formats. The following formats are supported:

- HTML
- PDF
- RTF
- MHTML

23.4 Reports for Oracle Identity Manager

All the reports containing Date type input parameters must be provided with the date range in the Date Input Parameters before running the report. Otherwise, the reports will not display any data.

Oracle Identity Manager Reports are classified into the following categories based on their functional areas:

- Access Policy Reports
- Request and Approval Reports
- Role and Organization Reports
- Password Reports
- Resource and Entitlement Reports
- User Reports
- Certification Reports
- Identity Audit Reports
- Exception Reports

23.4.1 Access Policy Reports

Oracle Identity Manager BI Publisher Reports provides the following access policy reports for Oracle Identity Manager:

- Access Policy Details
- Access Policy List by Role

23.4.1.1 Access Policy Details

It provides administrators or auditors the ability to view a current snapshot of all the policies defined in Oracle Identity Manager system, along with key information about each policy, and the number of instances in which each policy has been activated.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Access Policy Name	Name of the Access Policy

Fields

The following table lists the fields of the report:

Report Field	Description
Description	Description of the policy
Approval Required	Approval required for the policy
Creation Date	Date when the policy is created
Retrofit Access Policy	Retrofit of the access policy
Created By	Name of the person who created the policy
Priority	Priority of the policy

Columns

The following table lists the columns of the report:

Report Column	Description
Resource Name	Name of the resource

23.4.1.2 Access Policy List by Role

It lists all policies defined in Oracle Identity Manager system by role. This report can be used for operational and compliance purposes.

Input Parameters

The following table lists the input parameters for the report:

Report Parameter	Description
Role Name	Name of the role

Fields

The following table lists the fields of the report:

Report Field	Description
Description	Description of the policy
Approval Required	Approval required for the policy
Creation Date	Date when the policy is created
Retrofit Access Policy	Retrofit of the access policy
Created By	Name of the person who created the policy
Priority	Priority of the policy

Columns

The following table lists the columns of the report:

Report Column	Description
Role Name	Name of the role

23.4.2 Request and Approval Reports

Oracle Identity Manager BI Publisher Reports provides the following request and approval reports for Oracle Identity Manager:

- Approval Activity
- Request Details
- Request Summary
- Task Assignment History

23.4.2.1 Approval Activity

This report provides the administrators the ability to view the approval activity including requests that are approved, rejected, or pending.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Approver's First Name	First name of the approver

Report Parameter	Description
Approver's Last Name	Last name of the approver
Approver's User ID	User ID of the approver
Organization	Name of the organization

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
Approver's First Name	First name of the approver
Approver's Last Name	Last name of the approver
Approver's User ID	User ID of the approver
Organization	Organization of the approver
Approval Accepted	Count of the accepted approval
Approval Rejected	Count of the rejected approval
Approvals Pending	Count of the pending approval
Approval Requests Total	Total number of approval requests

23.4.2.2 Request Details

This report provides administrators the ability to view the details (requestor, current approver and so on) of all requests with the input current status. Additionally, this report displays the details of all users (user name, organization, manager details, user status and so on) that are provisioned as a result of the request approval. This helps administrators in planning and prioritizing operational activities so that they may expedite the closure of pending requests.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Requestor User First Name	First name of the requestor
Requestor User Last Name	Last name of the requestor

Report Parameter	Description
Request User ID	ID of the requestor
Request ID	Request ID
Request Parent ID	Parent ID of the request
Request Status	Status of the request
Request Type	Type of the request
Request Date From	Start date of the request
Request Date To	End date of the request
Beneficiary User First Name	First name of the beneficiary
Beneficiary User Last Name	Last name of the beneficiary
Beneficiary User ID	ID of the beneficiary

Fields

The following table lists the fields of the report:

Report Field	Description
Request ID	Request ID
Request Type	Type of the request
Requester User ID	ID of the requester
Request Date	Date on which request is initiated
Approver User ID	ID of the approver
Current Status	Status of the request
Parent Request ID	ID of the parent Requester

Columns

The following table lists the columns of the report, if a beneficiary is present:

Report Column	Description
First Name	First name of the beneficiary
Last Name	Last name of the beneficiary
User ID	ID of the beneficiary
User Type	Type of user
User Status	Status of the beneficiary

Report Column	Description
Organization	Organization of the beneficiary
Request Value	Request value of the resource

The following table lists the columns of the report, if a beneficiary is not present:

Report Column	Description
Request Name	Name of the request
Request Value	Value of the request

The following table provides the approver details:

Report Column	Description
Approver User ID	User ID of the approver of the request
Approver User Name	User name of the approver of the request

23.4.2.3 Request Summary

This report provides administrators the ability to view the current status of all requests raised in the specified time interval. This helps administrators in planning and prioritizing operational activities so that they may expedite the closure of pending requests.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Request Type	Type of request
Request Date From	Start date of the request
Request Date To	End date of the request
Organization	Details of the organization

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
Request ID	Request ID

Report Column	Description
Parent Request ID	ID of the parent Requester
Request Type	Type of request
Request Status	Status of request
Requester User ID	ID of the requester
Requester User Name	Name of the requester of the request
Beneficiary User ID	ID of the beneficiary
Request Details	Details of the request
Approver User ID	ID of the approver
Approver User Name	Name of the approver of the request
Request Date	Date of request

23.4.2.4 Task Assignment History

It lists the history of all task assignments.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
Assignee User ID	ID of the assignee user
Assignee First Name	First name of the assignee user
Assignee Last Name	Last name of the assignee user

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Column	Description
User ID	ID of the beneficiary
Assignee First Name	First name of the assignee
Assignee Last Name	Last name of the assignee
Assignee User ID	ID of the assignee
Assignee Role Name	Role name of the assignee

Report Column	Description
Assignee User Name	User name of the assignee
Employee Type	Type of employee

23.4.3 Role and Organization Reports

Oracle Identity Manager provides the following role and organization reports:

- Role Membership History
- Role Membership Profile
- Role Membership
- Organization Details
- User Membership History

23.4.3.1 Role Membership History

This report displays membership history of all the roles. The report will not show indirect memberships.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Role Name	Name of the role
Role Category	Category of the role
Employee Type	Type of the employee: Full-Time, Part-Time, Temp, Intern, Consultant, Contractor
Employee Status	Status of the employee: Active, Disabled, Deleted, Disabled Until Start Date
Membership Status	Status of membership: Revoked, Active
Effective From	Role membership effective from date
Effective To	Role membership effective to date

Fields

The following table lists the fields of the report:

Report Field	Description
Created By	Name of the person who created the role

Report Field	Description
Creation Date	Date on which the role was created

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Employee Type	Type of employee
Employee Status	Status of the employee
Membership Status	Membership date of the user
Effective From	Membership start date of the user
Effective To	Membership end date of the user
Manager's First Name	First name of the manager
Manager's Last Name	Last name of the manager
Manager's User ID	ID of the manager
Updated By	Name of the user who updated the record

23.4.3.2 Role Membership Profile

This report shows number of users present for number of roles and the details of users belonging to count number of roles.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Organization	Organization of the user

Fields

The following table lists the fields of the report:

Report Field	Description
Membership in Number of Roles	Number of members in number of roles
Number of Users	Number of users in the role

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Employee Type	Type of the employee: Full-Time, Part-Time, Temp, Intern, Consultant, Contractor

23.4.3.3 Role Membership

This report displays membership details of all roles.

Input Parameters

The following table lists input parameters for the report.

Report Parameter	Description
Role Name	Name of the role
Role Category	Category of the role
Organization	Name of the organization
Employee Type	Type of the employee: Full-Time, Part-Time, Temp, Intern, Consultant, Contractor
Employee Status	Status of the employee: Active, Disabled, Deleted, Disabled Until Start Date

Fields

The following table lists the fields of the report:

Report Field	Description
Created By	Name of the person who created the user
Creation Date	Date on which the user is created

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of user
Employee Status	Status of the user
Employee Type	Type of the employee: Full-Time, Part-Time, Temp, Intern, Consultant, Contractor
Member Since	Joining date of the user
Manager's First Name	First name of the manager
Manager's Last Name	Last name of the manager
Manager's User ID	ID of the manager

23.4.3.4 Organization Details

It lists the hierarchical organization structure and details about users in the organization.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Organization Name	Name of the organization

Fields

The following table lists the fields of the report:

Report Field	Description
Parent Organization Name	Name of the parent organization

Columns

The following table lists the columns of the report:

Report Column	Description
Role	Name of Administrator User roles
First Name	First name of the user in the organization

Report Column	Description
Last Name	Last name of the user in the organization
User ID	ID of the user
User Status	Status of the user
User Type	Type of user
Start Date	Joining date of the user
End Date	Leaving date of the user

23.4.3.5 User Membership History

This report lists the logged in users with their membership history.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Last Name	First name of the user
First Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Status	Status of the employee: Active, Disabled, Deleted, Disabled Until Start Date
Employee Type	Type of the employee: Full-Time, Part-Time, Temp, Intern, Consultant, Contractor

Fields

The following table lists the fields of the report:

Report Field	Description
User ID	ID of the user
User First Name	First name of the user
User Last Name	Last name of the user
Organization	Organization of the user
Employee Status	Status of the employee: Active, Disabled, Deleted, Disabled Until Start Date

Report Field	Description
Employee Type	Type of the employee: Full-Time, Part-Time, Temp, Intern, Consultant, Contractor

Columns

The following table lists the columns of the report:

Report Column	Description
User Role	Name of the user role
Membership Status	Status of membership
Effective From	Date from which the membership is effective
Updated By	User who updated the record

23.4.4 Password Reports

Oracle Identity Manager provides the following password reports:

- Password Expiration Summary
- Password Reset Summary
- Resource Password Expiration

23.4.4.1 Password Expiration Summary

This report shows the list of all active users whose Oracle Identity Manager passwords are about to expire within a specified period.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Last Name	Last name of the user
First Name	First name of the user
User ID	ID of the user
Organization	Organization of the user
Expiration Date Range From	Start date of the expiration date
Expiration Date Range To	End date of the expiration date

Fields

N/A

Columns

The following table lists the columns of the report:

Report Field	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Employee Type	Type of the employee: Full-Time, Part-Time, Temp, Intern, Consultant, Contractor
Employee Status	Status of the employee: Active, Disabled, Deleted, Disabled Until Start Date
Organization	Organization of the user
Password Expiration Date	Date on which the password expires

23.4.4.2 Password Reset Summary

This report provides the ability to view the aggregated metrics around password change attempts done by users themselves or on behalf of them. The metrics include all password change attempts, successful or failure outcome of password change attempt, users locked due to multiple concurrent unsuccessful password change attempts.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Aggregation Frequency	The frequency of the report generated
Date Range From	Start date of the report generated
Date Range To	End date of the report generated
Organization	Name of the organization

Fields

The following table lists the fields of the report:

Report Field	Description
Aggregation Frequency	The frequency of the report generated

Columns

The following table lists the columns of the report:

Report Column	Description
Time Period	Date and time of reset attempts performed
Reset Attempts	Number of reset attempts
Failed Reset Attempts	Number of failed reset attempts
Locked Users due to Failed Reset Attempts	Number of users locked due to a failed reset attempt
Resets by non-beneficiary	Number of resets by non-beneficiary

23.4.4.3 Resource Password Expiration

It lists users whose resource passwords will expire in a specified time period.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
User Status	Status of the user
Password Expiration Date From	The password expiry starting date
Password Expiration Date To	The password expiry ending date

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Field	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user

Report Field	Description
Organization	Organization of the user
User Status	Status of the user: Active, Disabled, Deleted, Disabled Until Start Date
User Type	Type of the user: Full-Time, Part-Time, Temp, Intern, Consultant, Contractor
Password Expiration Date	Date on which the password expires

23.4.5 Resource and Entitlement Reports

Oracle Identity Manager BI Publisher Reports provides the following resource and entitlement reports for Oracle Identity Manager:

- Account Activity In Resource
- Delegated Admins and Permissions by Resource
- Delegated Admins by Resource
- Entitlement Access List
- Entitlement Access List History
- Financially Significant Resource Details
- Resource Access List History
- Resource Access List
- Resource Account Summary
- Resource Activity Summary
- User Resource Access History
- User Resource Access
- User Resource Entitlement
- User Resource Entitlement History

23.4.5.1 Account Activity In Resource

It lists all account activities in each resource. It also provides information on how each user is associated with a specific activity of that resource.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
Date Range From	Date from which reports are displayed

Report Parameter	Description
Date Range To	Date to which reports are displayed

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Name	Name of the resource
Activity Type	The type of activity
Resource Authorizer User Role(s)	Name of the role which authorize the role
Resource Administrator User Role(s)	Name of the role which authorize the resource

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
User Status	Status of the user: Active, Disabled, Deleted, Disabled Until Start Date
Organization	Organization of the user
Manager's User ID	ID of the manager
Timestamp	Date when the report is created

23.4.5.2 Delegated Admins and Permissions by Resource

This report displays the list of user roles with write and delete access that are administrators of the resource.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
Administrator Role Name	Name of the Administrator role
Administrator Role Information	Information about the Administrator role
Read Access	Indicates whether the resource has read access
Write Access	Indicates whether the resource has write access
Delete Access	Indicates whether the resource has delete access
Authorizer Role	Authorizer role name
Name Priority	Priority of the resource
Created By	Name of the person who created the resource
Creation Date	Resource creation date

23.4.5.3 Delegated Admins by Resource

The report displays the list of user roles that are the administrators or authorizers of the resource and members of those roles.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
Resource Type	Type of resource
Resource Audit Objective	Objective to carry out the audit for the resource

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource

Report Field	Description
Target	Indicates whether the resource is a target for organization or user
Write Access	Indicates whether the resource has write access
Delete Access	Indicates whether the resource has delete access
Creation By	Resource creation source
Creation Date	Date on which resource is created

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
User Status	Status of the user
Member Since	Joining date of the user
Manager's First Name	First name of the manager
Manager's Last Name	Last name of the manager
Manager's User ID	ID of the manager

23.4.5.4 Entitlement Access List

This report provides administrators or auditors the ability to query all existing users, who have a specified entitlement. This report can be used for operational and compliance purposes.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Entitlement Code	Code of the entitlement
Resource Name	Name of the resource
Organization	Organization of the user

Report Parameter	Description
Role Name	Name of the role
User Status	Status of the user: Active, Disabled, Deleted, Disabled Until Start Date
User Type	Type of user
Provisioning Date From	Date from which the resource is provisioned to the user
Provisioning Date To	Date to which the resource is provisioned to the user

Fields

The following table lists the fields of the report:

Report Field	Description
Entitlement Code	Code of the entitlement
Entitlement Name	Name of the entitlement
Entitlement status	Status of the entitlement.
Resource Name	Name of the resource
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Column	Description
User Id	ID of the user
First Name	First name of the user
Last Name	Last name of the user
User Status	User Status
User Type	Type of the user
Organization	Organization of the user
Valid To Date	Entitlement valid from date
Valid From Date	Entitlement valid to date

23.4.5.5 Entitlement Access List History

This report provides administrators or auditors the ability to query all existing users provisioned to a entitlement over its lifecycle. This is a lifetime report showing entire history of resource's access list or entitlements.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Entitlement Code	Code of the entitlement
Resource Name	Name of the resource
Organization	Organization of the user
Role Name	Name of the role
User Status	Status of the user: Active, Disabled, Deleted, Disabled Until Start Date
User Type	Type of user
Effective From Date	Entitlement effective from date
Effective To Date	Entitlement effective to date

Fields

The following table lists the fields of the report:

Report Field	Description
Entitlement Code	Code of the entitlement
Entitlement Name	Name of the entitlement
Resource Name	Name of the resource
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Column	Description
User Id	ID of the user
First Name	First name of the user
Last Name	Last name of the user
User Status	Status of the user
User Type	Type of user

Report Column	Description
Effective From	Entitlement effective from date
Effective To	Entitlement effective to date

23.4.5.6 Financially Significant Resource Details

This report provides Administrators to get a list of financially significant resources to prioritize various administrative and cleanup activities. It also helps Compliance or Privacy and Security officers assessing effectiveness of preventive and detective controls in financial significant resources and Auditors to understand the IT resources that host financial data.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Column	Description
User Roles	Lists the resource administrator user roles

23.4.5.7 Resource Access List History

This report provides administrators or auditors the ability to query all existing users provisioned to a resource over its lifecycle. This is a lifetime report showing entire history of resource's access list or entitlements.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user

Report Parameter	Description
Organization	Organization of the user
User Status	Status of the user
User Type	Type of the user
Snapshot Date From	Effective start date of resource access to the user
Snapshot Date To	Effective end date of resource access to the user
Changes Date From	Resource changed from date to user
Changes Date To	Resource changed to date to user

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Resource Descriptive data	Descriptive data to identify the resource
User Status	Status of the user
Resource Status	Status of the resource
Effective From	Effective start date
Effective To	Effective end date

23.4.5.8 Resource Access List

This report provides administrators or auditors the ability to query all existing users provisioned to a specified resource. This report can be used for operational and compliance purposes.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
User Status	Status of the user
User Type	Type of the user
Provisioning Date From	Resource provision start date
Provisioning Date To	Resource provision end date

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
User Type	Type of the user
User Status	Status of the user
Organization	Organization of the user
Provisioning Date	Date on which the resource is provisioned

23.4.5.9 Resource Account Summary

This report lists the number of users for each status within each resource.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
Resource Type	Type of resource
Account Status	Status of the account

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource
Total Number of Users	Total number of users associated with the account

Columns

The following table lists the columns of the report:

Report Column	Description
Account Status	Status of the account
Number of Users	Number of users with that account status

23.4.5.10 Resource Activity Summary

It lists the history of all provisioning and approval activities for a resource.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
Date Range From	Start date
Date Range To	End date

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Column	Description
Accounts Provisioned	Number of accounts provisioned
Accounts De-Provisioned	Number of accounts de-provisioned
Approval Requests	Number of approval requests

Report Column	Description
Approval Accepted	Number of approved requests
Approval Rejected	Number of rejected requests

23.4.5.11 User Resource Access History

This report provides administrators or auditors the ability to view user's resource access history over user's lifecycle. This report can be used for compliance and forensic auditing purposes. This is not a user access profile snapshot report. This is a lifetime report showing entire history of user's entitlements.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Status	Status of the user
Employee Type	Type of employee

Fields

The following table lists the fields of the report:

Report Field	Description
User ID	ID of the user
User First Name	First name of the user
User Last Name	Last name of the user
Manager User ID	ID of the reporting Manager
Manager First Name	First name of the reporting Manager
Manager Last Name	Last name of the reporting Manager
Organization	Organization of the user
Employee Status	Status of employee
Employee Type	Type of employee
Identity Creation Date	User creation date

Columns

The following table lists the columns of the report:

Report Column	Description
Resource Name	Name of the resource
Resource Descriptive Data	Description of the resource
Provisioned Date	Date on which the resource is provisioned
Provisioned By	Name of the person who provisioned the resource
Effective From	Effective start date of resource access to the user
Effective To	Effective end date of resource access to the user

23.4.5.12 User Resource Access

This report provides administrators or auditors the ability to query all existing users provisioned to a specified resource. This report can be used for operational and compliance purposes.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Status	Status of employee
Employee Type	Type of employee

Fields

The following table lists the fields of the report:

Report Field	Description
User ID	ID of the user
User First Name	First name of the user
User Last Name	Last name of the user
Manager User ID	ID of the reporting Manager
Manager First Name	First name of the reporting Manager
Manager Last Name	Last name of the reporting Manager

Report Field	Description
Organization	Organization of the user
Employee Status	Status of employee
Employee Type	Type of employee
Identity Creation Date	User creation date

Columns

The following table lists the columns of the report:

Report Column	Description
Resource Name	Name of the resource
Resource Descriptive Data	Description of the resource
Resource Status	Status of the resource
Provisioned Date	Date on which the resource is provisioned

23.4.5.13 User Resource Entitlement

This report provides administrators or auditors the ability to query all existing entitlements provisioned to specific users. This report can be used for operational and compliance purposes.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
User ID	ID of the user
First Name	First name of the user
Last Name	Last name of the user
Email	Email of the user
Resource Name	Name of the resource
Organization	Organization of the user
Role Name	Name of the role
User Status	Status of the user
User Type	Type of the user

Fields

The following table lists the fields of the report:

Report Field	Description
User ID	ID of the user

Report Field	Description
First Name	First name of the user
Middle Name	Middle name of the user
Last Name	Last name of the user
Email	Email of the user
Organization	Organization of the user
User Status	Status of the user
User Type	Type of the user
Manager First Name	First name of the manager
Manager Last Name	Last name of the manager
Start Date	Entitlement of resource start date
End Date	Entitlement of resource end date

Columns

The following table lists the columns of the report:

Report Column	Description
Entitlement Code	Code of the entitlement
Entitlement Name	Name of the entitlement
Entitlement Status	Status of the entitlement
Resource	Type of the resource
Provisioning Start	Date from which the resource is provisioned to the user
Valid From Date	Entitlement of resource valid start date

23.4.5.14 User Resource Entitlement History

This report provides administrators or auditors the ability to view user's resource entitlement history over user's lifecycle. This report can be used for compliance and forensic auditing purposes. This is not a user access profile snapshot report. This is a lifetime report showing entire history of user's entitlements.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
User ID	ID of the user
First Name	First name of the user
Last Name	Last name of the user
Email	Email of the user
Resource Name	Name of the resource
Organization	Organization of the user
Role Name	Name of the role
User Status	Status of the user
User Type	Type of the user
Effective From Date	Resource entitlement effective start date
Effective To Date	Resource entitlement effective end date

Fields

The following table lists the fields of the report:

Report Field	Description
User ID	ID of the user
First Name	First name of the user
Last Name	Last name of the user
User Status	Status of the user
User Type	Type of the user
Organization	Organization of the user
Email	Email of the user
Start Date	Start date of resource entitlement
End Date	End date of resource entitlement
Identity Creation Date	Date of identity creation
Manager First Name	First name of the manager
Manager Last Name	Last name of the manager

Columns

The following table lists the columns of the report:

Report Column	Description
Entitlement Code	Code of the entitlement
Entitlement Name	Name of the entitlement
Resource	Type of the resource
Effective From Date	Resource entitlement effective start date
Effective To Date	Resource entitlement effective end date

23.4.6 User Reports

Oracle Identity Manager provides the following user reports:

- User Creation
- User Profile History
- User Summary
- Users Deleted
- Users Disabled
- Users Unlocked

23.4.6.1 User Creation

This report lists all Oracle Identity Manager users created between a specified date range. In addition, it provides the source of information on the users created.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Employee Status	Status of the user
Employee Type	Type of employee
Creation Date From	Start date of user summary
Creation Date To	End date of user summary
Organization	Organization of the user

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Current Status	Status of the user
Employee Type	Type of employee
Manager ID	ID of the Manager to whom the user reports
Source of Creation	User creation source
Creation On	Date on which the user is created
Created By	User who created the user

23.4.6.2 User Profile History

This report shows all the users and their details based on the input parameters.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Role Name	Role of the user
Manager User ID	ID of the Manager to whom the user reports
Employee Status	Status of the user
Employee Type	Type of employee
Changes Date Range From	Effective start date of the changes
Changes Date Range To	Effective end date of the changes
Snapshot Date Range From	Effective start date of resource access to the user
Snapshot Date Range To	Effective end date of resource access to the user

Fields

The following table lists the fields of the report:

Report Field	Description
User ID	ID of the user
User First Name	First name of the user
User Last Name	Last name of the user
Manager User ID	ID of the reporting Manager
Manager First Name	First name of the reporting Manager
Manager Last Name	Last name of the reporting Manager
Organization	Organization of the user
Employee Status	Status of employee
Employee Type	Type of employee
Identity Creation Date	User creation date

Columns

The following table lists the columns of the report:

Report Column	Description
Profile Parameter	Name of user profile
Value	Value of user profile
Date Effective From	Effective from date
Time Effective From	Effective from time
Updated By	User who updated the record

23.4.6.3 User Summary

It lists all Oracle Identity Manager User's summary in a specified time period. It includes user details along with source of creation, and who created it and when.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Employee Status	Status of the user
Employee Type	Type of employee
Creation Date From	Start date of user summary

Report Parameter	Description
Creation Date To	End date of user summary
Organization	Organization of the user

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Status	Status of the user
Employee Type	Type of employee
Manager ID	ID of the Manager to whom the user reports
Source	User creation source
Creation Date	Date at which the user is created

23.4.6.4 Users Deleted

This report shows all the deleted users and their details based on input parameters.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Type	Type of employee
Deletion Date From	Start date of summary of deleted users
Deletion Date To	End date of summary of deleted users
Organization	Organization of the user

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Type	Type of employee
Manager ID	ID of the Manager to whom the user reports
Source	User creation source
Deletion Date	Date at which the user is deleted

23.4.6.5 Users Disabled

This report provides the ability to view the details of users whose accounts are disabled. The account may be disabled for various reasons, for example, unsuccessful login or password reset attempts failure.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Type	Type of employee
Disabled Date From	Start date of user disabled
Disabled Date To	End date of user disabled
Organization	Organization of the user

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user

Report Column	Description
Organization	Organization of the user
Employee Status	Current status of the employee
Employee Type	Type of employee
Manager ID	ID of the Manager to whom the user reports
Source	User creation source
Disabled Date	Date at which the user is disabled
Updated By	Users who updated the record

23.4.6.6 Users Unlocked

This report provides the ability to view the details of users whose disabled accounts are unlocked by administrators. Delegated administrators of the organizations to whom the user belongs may enable the accounts.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Type	Type of employee
Unlocked Date From	Start date of user unlocked
Unlocked Date To	End date of user unlocked
Organization	Organization of the user

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
Employee Status	Status of the user

Report Column	Description
Employee Type	Type of employee
Manager ID	ID of the Manager to whom the user reports
Source	User creation source
Unlocked Date	Date at which the user is unlocked
Updated By	User who updated the record

23.4.7 Certification Reports

Certification reports select data from the certification tables of the Oracle Identity Manager database. There are a list of predefined or default certification reports in Oracle Identity Manager. Table 23–1 lists the default certification reports for each type of certification.

Table 23–1 Default Certification Reports

Certification Type	Certification Report	Description
User certification	Complete Certification Report	Presents comprehensive data of a user certification. This report includes a list of all employees and their access.
	Certified Access Report	Lists access marked as certified.
	Revoked Access Report	Lists access marked as revoked.
	Abstained Access Report	Lists certification items that the certifier declined to complete because the certifier is not responsible for verifying the user's assigned roles and entitlements.
	Certified Conditionally Access Report	Lists access that the certifier approved temporarily, even though the access may not be appropriate or justified in the long term. Reviewers are required to enter an end date, which is included in this report. However, the access is not revoked and notices are not sent out about expired end dates.
	Complete Certification Task Report	Presents user-certification data based on certification tasks. This is a subset of the Complete Certification Report.
Role certification	Complete Certification Report	Presents comprehensive data of a role certification.
	Certified Access Report	Lists entitlements marked as certified.
	Revoked Access Report	Lists entitlements as revoked.
	Abstained Access Report	Lists certification items that the certifier declined to complete because the certifier is not responsible for verifying the role's assigned memberships.
	Certified Conditionally Access Report	Lists access that the certifier approved temporarily, even though the access may not be appropriate or justified in the long term. Reviewers are required to enter an end date, which is included in this report. However, the access is not revoked and notices are not sent out about expired end dates.
	Complete Certification Task Report	Presents role-certification data based on certification tasks. This is a subset of the Complete Certification Report.

Table 23–1 (Cont.) Default Certification Reports

Certification Type	Certification Report	Description
Application instance certification	Complete Certification Report	Presents comprehensive data of an application instance certification.
	Certified Access Report	Lists entitlements marked as certified.
	Revoked Access Report	Lists entitlements marked as revoked.
	Abstained Access Report	Lists certification items that the certifier declined to complete because the certifier is not responsible for verifying the application instances's assigned users and accounts.
	Certified Conditionally Access Report	Lists access that the certifier approved temporarily, even though the access may not be appropriate or justified in the long term. Reviewers are required to enter an end date, which is included in this report. However, the access is not revoked and notices are not sent out about expired end dates.
	Complete Certification Task Report	Presents certification data for application instances based on certification tasks. This is a subset of the Complete Certification Report.
Entitlement certification	Complete Certification Report	Presents comprehensive data of an entitlement certification.
	Certified Access Report	Lists access marked as certified.
	Revoked Access Report	Lists access marked as revoked.
	Abstained Access Report	Lists certification items that the certifier declined to complete because the certifier is not responsible for verifying the entitlement's assigned accounts and attributes.
	Certified Conditionally Access Report	Lists access that the certifier approved temporarily, even though the access may not be appropriate or justified in the long term. Reviewers are required to enter an end date, which is included in this report. However, the access is not revoked and notices are not sent out about expired end dates.
	Complete Certification Task Report	Presents entitlement-certification data based on certification tasks. This is a subset of the Complete Certification Report.

23.4.8 Identity Audit Reports

IDA Policy Violation Reports are available for download from Reports link in the **Compliance** tab of Oracle Identity Self Service. An IDA Policy Violation report can be generated for a Policy, Scan Stop Date, Manager, Remediator or selected users.

The following types of reports are available:

- **Closed Policy Violation Report:** Contains all the policy violations that are in Closed state.
- **Remediation Completed Policy Violation Report:** Contains all the policy violations that are in Remediation Completed state.
- **Expired Policy Violation Report:** Contains all the policy violations that are in Expired state.
- **Remediation In Progress Policy Violation Report:** Contains all the policy violations that are in Remediation In Progress state.

- **Remediation Under Review Policy Violation Report:** Contains all the policy violations that are in Remediation Under Review state.
- **Open Policy Violation Report:** Contains all the policy violations that are in Open state.
- **Preview Policy Violation Report:** Contains all the policy violations that are in Preview state.
- **Assigned Policy Violation Report:** Contains all the policy violations that are in Assigned state.

23.4.9 Exception Reports

In Oracle Identity Manager, **exception** refers to the difference between accounts that a user is entitled to and the accounts that are actually assigned to a user. The user is assigned these accounts as a result of access policies, provisioning of resources, approval requests, and reconciliation events. Any difference of these accounts assigned to a user in the target system and the ones assigned to the user in Oracle Identity Manager comprises an exception. Exception reports are enabled by default.

Oracle Identity Manager provides the following exception reports:

- **Rogue Accounts By Resource**

  This report returns a list of all the rogue accounts existing in a resource. The following exceptions are reported:

  - Account exists in the target system, but has been deprovisioned for the corresponding user in Oracle Identity Manager
  - Account exists and is active in the target system, but account does not exist in Oracle Identity Manager (user exists)
  - Account exists and is active in the target system, but user does not exist in Oracle Identity Manager
  - Account exists and is active in the target system, but Oracle Identity Manager user has been disabled
  - Account exists and is active in the system target, but Oracle Identity Manager user has been deleted

- **Orphaned Account Summary Report:** An account that exists in the target system, but the corresponding user to whom the account is provisioned has been deleted in Oracle Identity Manager. For the given input resource, it lists the rogue accounts that exist in the target system, but the corresponding users to whom the accounts are provisioned has never existed in Oracle Identity Manager.
- **Fine Grained Entitlement Exceptions By Resource**

  This report returns a list of all the accounts in a resource for which the process form data being reconciled is different from the expected values. It means that this report returns any account existing in the target system that is also provisioned to the corresponding user in Oracle Identity Manager, but for which the process data does not match.

Note:

- After completion of initial target reconciliation, all account-related activities performed directly on a target resource are tracked as exception activity. Account-related activities include account creation, account modification, and entitlement assignment/revocation. The exception reports should be used only if the organization policies enforce that all account-related activities in target resources would always be initiated in Oracle Identity Manager. In addition, remember that exception detection and recording are an extension of account data reconciliation and, therefore, may result in a drop in performance during reconciliation.
- All the exception reports depend on reconciliation data. Therefore, these reports will not display any data if the corresponding reconciliation events are archived.

This section describes the following exception reports:

- Fine Grained Entitlement Exceptions By Resource
- Orphaned Account Summary
- Rogue Accounts By Resource

23.4.9.1 Fine Grained Entitlement Exceptions By Resource

This report enables administrators, signing officers, internal and external auditors to analyze discrepancies in various process forms and related child tables of various resources and mitigate material weaknesses in the resources through remediation activities.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Employee Type	Type of the employee such as fulltime, part time
Organization Name	Name of the organization
Role Name	Name of the role

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Name	Name of the resource
User ID	ID of the user
First Name	First name of the user
Last Name	Last name of the use
Organization Name	Name of the organization
Employee Status	Status of the user
Employee Type	Type of the user
Reviewer First Name	First name of the reviewer
Reviewer Last Name	Last name of the reviewer
Unique ID Attribute	Unique ID attribute in account profile
Unique ID Value in Account Profile	Unique ID value in account profile

Columns

The following table lists the columns of the report:

Report Column	Description
Form Name	Name of the form
Form Type	Type of the form
Form Field Name	Field name of the form
Expected Form Field Value	Old value of the field
Actual Form Field Value	New value of the field

Note: Before running this report, you must populate data for account audit and reconciliation exceptions.

To populate the data for account audit and reconciliation exceptions:

1. Provision an user to any target.
2. Modify any of the user's attribute in the target and reconcile the user.
3. Find data in UPA_UD_FORMFIELDS and UPA_UD_FORMS tables.
4. Go to Oracle Identity Manger server and run RefreshMaterializedViewScheduler Task.
5. Log in to BIP and view the report.

23.4.9.2 Orphaned Account Summary

It lists the rogue accounts for the input resource for which a user existed in the target system, but the associated user to whom the account is provisioned never existed in Oracle Identity Manager.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
Reconciliation Date Range From	Start date of reconciliation
Reconciliation Date Range To	End date of reconciliation

Fields

N/A

Columns

The following table lists the columns of the report:

Report Column	Description
Resource	Name of the resource
Account Information	Information of the orphaned account
Account Detail	Details of the account associated with this orphaned account
Reconciliation Date	Date of reconciliation

23.4.9.3 Rogue Accounts By Resource

This report includes all rogue accounts for the input resource. This enables administrators, signing officers, internal and external auditors to identify material weaknesses in the resources and plan their mitigation through remediation activities.

Input Parameters

The following table lists the input parameters for the report.

Report Parameter	Description
Resource Name	Name of the resource
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization Name	Organization of the user
User Status	Status of the user
User Type	Type of the user
Exception Type	Type of exception

Fields

The following table lists the fields of the report:

Report Field	Description
Resource Type	Type of resource

Columns

The following table lists the columns of the report:

Report Column	Description
Exception Type	Type of exception
First Name	First name of the user
Last Name	Last name of the user
User ID	ID of the user
Organization	Organization of the user
User Status	Status of the user
User Type	Type of the user
Account Details	Details of the rogue account
Reviewer First Name	First name of the reviewer
Reviewer Last Name	Last name of the reviewer
Reviewer User ID	User ID of the reviewer

23.5 Required Scheduled Tasks for BI Publisher Reports

Table 23–2 lists the scheduled tasks required for Oracle Identity Manager BI Publisher reports:

Table 23–2 Scheduled Tasks for BI Publisher Reports

Report Name	Scheduled Task Name	Description
Fine Grained Entitlement Exceptions By Resource	RefreshMaterializedView	To refresh the Materialized View used in this report with the latest data
User Profile History	IssueAuditTask	To populate the audit tables with the latest data
User Unlocked	IssueAuditTask	To populate the audit tables with the latest data
User Membership History	IssueAuditTask	To populate the audit tables with the latest data
Role Membership History	IssueAuditTask	To populate the audit tables with the latest data
Resource Access List History	IssueAuditTask	To populate the audit tables with the latest data
User Resource Access History	IssueAuditTask	To populate the audit tables with the latest data

Table 23–2 (Cont.) Scheduled Tasks for BI Publisher Reports

Report Name	Scheduled Task Name	Description
Resource Activity Summary	IssueAuditTask	To populate the audit tables with the latest data
Password Reset Summary	IssueAuditTask	To populate the audit tables with the latest data
Entitlement Reports	Entitlement List	To populate the Entitlement List table with the marked entitlements
	Entitlement Assignment	To populate the Entitlement Assignment tables with the assigned entitlements
	Entitlement Updates	To populate the latest data into the Entitlement Assignment tables, if any entitlement has assigned to any user periodically or later

23.6 Best Practices for Running Oracle Identity Manager Reports

As a best practice, you must consider the following points before running Oracle Identity Manager BI publisher reports:

- Do not run Oracle Identity Manager reports with null value in date range parameters. You must run Oracle Identity Manager reports always with date range values in data range parameters, otherwise report will not display anything.
- Invoke the reports with the set of values as input parameters to provide the selectivity, thus improving the performance.
- By default, the System Administrator user of Oracle Identity Manager has all the permissions to login to BI Publisher and access all the Oracle Identity Manager Reports.

24

Using the Archival and Purge Utilities for Controlling Data Growth

The application capabilities in Oracle Identity Manager generate a large volume of data. To meet the standards of performance and scalability, maintaining the data generated for the life cycle management of Oracle Identity Manager entities is a challenge. Oracle Identity Manager meets this challenge by providing online and continuous as well as offline data purge and archival solutions.

Table 24–1 lists the archival and purge solutions provided by Oracle Identity Manager for its entities and their dependent data.

Table 24–1 Archival and Purge Solutions

Archival and Purge for Entities	Real-time Online Mode	Operated via Command Line	Available via Other Modes
Reconciliation	Yes	Yes	
Provisioning Tasks	Yes	Yes	
Request	Yes	Yes	
Orchestration	Yes	No	
Lightweight Audit	Yes	No	For more information on Partition based Approach, see "Audit Data Growth Control Measures in Lightweight Audit Framework" on page 24-24.
Legacy Audit	No	No	For more information on Partition based Approach, see "Audit Data Growth Control Measures in Legacy Audit Framework" on page 24-27.
Certification	Yes	No	

Note: Archival and purge solution for the **certification entity** is available only if you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170418. For instructions on downloading and applying the 11.1.2.3.161018 bundle patch, refer to the bundle patch documentation.

This chapter describes how to use the various archival and purge utilities and the concepts related to them. It contains the following topics:

- Understanding Archival and Purge Concepts
- Using Real-Time Purge and Archival Option in Oracle Identity Manager
- Using Command-Line Option of the Archival Purge Utilities in Oracle Identity Manager
- Using the Audit Archival and Purge Utility
- Using the Real-Time Certification Purge in Oracle Identity Manager

Note:

- Oracle recommends that you use the real-time purge and archival option rather than the command-line utilities.
- The archival and purge utilities (scheduled task-based and command-line) only purge data from the underlying Oracle Identity Manager tables, and do not reclaim space. For information about reclaiming space, see the document titled *How To Reclaim Space For Overgrown/Huge Footprint Of LOB Columns In OIM Database* (Doc ID 2017034.1) in the My Oracle Support web site at the following URL:

  `https://support.oracle.com`

24.1 Understanding Archival and Purge Concepts

The concepts related to archival and purge solutions in Oracle Identity Manager are described in the following sections.

- Categorization: Purge Only Solution Versus Purge and Archive Solution for Entities
- Archival of Data
- Purge
- Real-Time Purge
- Retention Period
- Modes of Archival Purge Operations

24.1.1 Categorization: Purge Only Solution Versus Purge and Archive Solution for Entities

The purge-only solution and the purge plus archive solution is applicable to the real-time purge and archival feature. Oracle Identity Manager entities are divided into

the following on the basis of how the data related to them are purged and archived from the perspective of Real-time Purge Archival feature:

- **Purge only:** Entities for which data is directly purged but not archived. These entities are Reconciliation, Provisioning Tasks, and Orchestration.
- **Purge and archive:** Entities for which data is purged as well as archived. This is applicable to the Request entity.

> **Note:** The real-time purge and archival solution provides data purge capabilities on a continuous basis. In addition, you can use the command-line archival utilities periodically to archive data, if required. There is no such categorization of entities in their command-line archive purge utilities version. They essentially archive prior to purge. For details about the command-line archival utilities, see "Using Command-Line Option of the Archival Purge Utilities in Oracle Identity Manager" on page 24-11.

24.1.2 Archival of Data

Archival (prior to purge) is the standard mechanism followed in Oracle Identity Manager command-line utilities that offer for deleting data from the Active Feature or Entity tables. This is done by copying the data to a shadow copy or replica of the original table, typically with a suffix ARCH_*TABLE_NAME*. Archive operation precedes purge in data purge solutions for entities in the Purge and Archive category.

24.1.3 Purge

The mechanism to delete or purge data from the Active Feature or Entity tables without any preceding archive operation. Data purged is non-recoverable in Oracle Identity Manager.

24.1.4 Real-Time Purge

Real-time purge denotes that data is deleted or purged when Oracle Identity Manager is up and running and is available irrespective of the feature invocation, concurrency, or workload. However, in contrast to the literal meaning of real-time, entity data created in the system is not deleted immediately.

24.1.5 Retention Period

Retention period defines the age of the data that needs to be retained in Oracle Identity Manager for functional usage and compliance purpose. Data is deleted based on the age defined by the retention period value for the entity data in question. The Retention Period attribute must be defined for the real-time purge feature via the OIM Data Purge scheduled job user interface.

24.1.6 Modes of Archival Purge Operations

Archival purge operations can be performed in the following modes:

- **Offline mode:** In this mode, archival and purge of data renders Oracle Identity Manager unusable for the time period it is being run. Because the entire operation being database-intensive, it disables the constraints/indexes at the beginning, copies, deletes the data from the entity tables, and re-enables the post deletion. This is for attaining the maximum performance in the delete operation and

eliminating possibilities of functional inconsistencies in the data entered in the window of deletion with table-level constraints disabled. Therefore, any transactional-level changes from Oracle Identity Manager usage is not advised, and as a result, the system is offline from the usage perspective.

- **Online mode:** In this mode, archival and purge of data happens with the entire database-level indexes/constraints enabled as usual. Therefore, Oracle Identity Manager usage can be continued in online mode from the operational perspective.

Note: Real-time purge supports online mode only. Command-line Archival Purge Utilities support both online and offline modes based on the user input.

24.2 Using Real-Time Purge and Archival Option in Oracle Identity Manager

The application capabilities in Oracle Identity Manager generate a large volume of data. To meet the standards of performance and scalability, maintaining the data generated for the life cycle management of Oracle Identity Manager entities is a challenge. Oracle Identity Manager meets this challenge by providing a real-time and continuous data purge solution, which is described in the following sections:

- Understanding Real-Time Data Purge and Archival
- Configuring Real-Time Purge and Archival
- Understanding the Orchestration Purge Utility
- Collecting Diagnostic Data of the Online Archival and Purge Operations

24.2.1 Understanding Real-Time Data Purge and Archival

The real-time purge and archival capability is provided by default in Oracle Identity Manager. Entity data can be continuously purged through this based on the options or choices made.

The configuration is one time and the purge solution works automatically without any intervention from the administrator.

The real-time purge and archival has the following features:

- The administrators provides values for some critical parameters, such as retention period, run duration, and purge criteria, for entities by using the Scheduled Tasks section of Oracle Identity System Administration.
- Diagnostic information about each purge run is captured as a log.
- Purge tasks run periodically.
- The entity modules, such as Request, Reconciliation, Task, and Orchestration, is purged according to the allotted time duration.
- The purge solution is fail safe. This means that in the event of a situation, the system does not endlessly consume CPU cycles. A fail-safe design has a minimum impact on other modules. The fail-safe capability is provided by:
  - Maximum Run Time for Auto-Cutoff in Purge Run for each Entity: Each run of the purge utility is governed by the value of the Maximum Purge Run Time parameter, the value of which is in minutes. Purge automatically stops when this maximum purge run duration is exceeded. This is provided at the each

entity level so that you can control the Purge Time Period allocation at the feature level.

Each batch picked up for deletion is aware of the time factor. When the time factor exceeds, the next batch is skipped and the utility's flow of control comes to completion.

The Maximum Purge Run Time in minutes for each entity can be specified in the scheduled task UI.

– Single-threaded batching: The purge operation accepts a batch size, which is the maximum number of rows to delete before a commit is issued. This keeps the redo log segments from growing too large when purge is applied to a large number of rows. The batch size is accepted from the scheduled task interface for the purge run operation.

- Data growth and subsequent footprint is controlled on an on-going basis.
- It operates online with no disruption of service.
- The purge operation via an automated scheduled task runs silently at a predefined periodicity and is non-interactive. Various metrices related to the purge operation, such as names of the entity modules, success or failure status, and number of rows targeted for deletion, are logged. These logs are diagnostic pointers for the purge operation for every run.
- The volume of data purged through the Real-time Purge Utilities Framework is a function of a few inputs, such as time duration window, entities selected, and existing workload.There might be instances when outflow of data in Oracle Identity Manager via this purge functionality is less than the inflow, which means that there would be some data volume accumulating in the system. This can be then purged via the Command-Line Archival/Purge Utilities at a reasonable point in time.

24.2.2 Configuring Real-Time Purge and Archival

Entity data via the Purge solution is continuously purged based on the options or choices that you make when you configure running of the utility. You can modify these options based on data retention policies and maintenance requirements.

To configure real-time purge and archival:

1. Log in to Oracle Identity System Administration.
2. Under System Management, click **Scheduler**.
3. Search and open the 'OIM Data Purge Job' scheduled job.
4. In the Parameters section, specify values for the parameters, as described in Table 24–2:

Table 24–2 Purge Configuration Parameters

Category	Parameter	Description	Default Value
Global parameters	Batch Size	The purge operation runs in batches. It represents the maximum number of rows to delete before a commit is issued.	5000
	Maximum Purge Run Duration Per Entity(in Mins)	This is the maximum run duration in minutes for purge processing for each entity.	30 mins
Orchestration purge parameters	Orchestration Entity Selection	This specifies whether or not data is to be purged from orchestration tables.	Yes
	Orchestration Purge Criteria	This takes the following values: ■ 1 for completed orchestrations ■ 2 for failed, compensated, canceled, or canceled with compensation orchestrations ■ 3 for both 1 and 2	1
	Orchestration[COMPLETED] Retention Period(in days)	This indicates the retention period in days for completed orchestrations.	1 day
	Orchestration[OTHERS] Retention Period(in days)	This indicates the retention period in days for failed, compensated, or other orchestrations	30 days
Provisioning task purge parameters	Provisioning Task Entity Selection	This specifies whether or not data is to be purged from provisioning task tables.	No
	Provisioning Tasks Purge Criteria	This takes the following values: ■ 1 for completed provisioning tasks ■ 2 for completed and canceled provisioning tasks	1
	Provisioning Tasks Retention Period(in days)	This indicates the retention period in days for provisioning tasks.	90 days
Reconciliation purge parameters	Recon Entity Selection	This specified whether or not data is to be purged from reconciliation tables.	No
	Recon Purge Criteria	This takes the following values: ■ 1 for completed reconciliation events ■ 2 for linked reconciliation events ■ 3 for both 1 and 2	1
	Recon Retention Period(in days)	This indicates the retention period in days for reconciliation events.	30 days
Request purge parameters	Request Entity Selection	This specifies whether or not data is to be purged from request tables.	No
	Request Purge Criteria	This takes the following values: ■ 1 for completed requests ■ 2 for failed requests ■ 3 for completed and failed requests	1
	Request Retention Period(in days)	This indicates the retention period in days for requests.	90 days

Note: By default, the 'OIM Data Purge Job' scheduled job is available in the enabled state with a retention period of 90 days. You must revisit the job parameters to disable or to change the purge interval as required.

5. Click **Apply**.

In addition to the steps on the Scheduled Task UI for configuration inputs documented in this section, there are no further steps required manually, such as archival tablespace creation. All the steps in the subsequent sections are for running the command-line version of the utilities.

Note:

- For Real-time Archival Purge operation via Scheduled Task interface, Retention Period must not be specified as ZERO as this can cause inconsistencies in purge operation.
- Simultaneous runs of multiple 'OIM Data Purge Job' scheduled jobs is not supported via instantiation of the Scheduled Task functionality.
- There should be no overlap of archival/purge utility run for an entity from both modes in Oracle Identity Manager, which are scheduled task and command-line modes.
- For details of the purge internals, such as tables that undergo purge for Request, Reconciliation, and Provisioning Tasks, refer to the subsequent sections of the command-line utilities. Both real-time scheduled job-based purge and command-line archival utilities purge data from the same set of table for an entity.
- If database is restarted when any scheduled job is running, then the job is stuck in RUNNING status. You need to restarting the scheduler service to stop all the jobs which are stuck in RUNNING status.

  For more information on how to stop the scheduled services, see Starting and Stopping the Scheduler.

24.2.3 Understanding the Orchestration Purge Utility

Orchestration data purge takes place from the active orchestration tables via the unified 'OIM Data Purge Job' scheduled job interface. It is based on the following criteria:

Note: Orchestration purge is available only in online mode and via the scheduled job interface.

- Orchestration process status, such as Completed, Failed, Compensated, Canceled, or Canceled with Compensation.
- Time-based criteria, which is specified via the retention period value specified in days on the scheduled job interface.

The following active orchestration tables undergo purge via the Orchestration Purge feature:

- ORCHPROCESS
- CALLBACK_INVOCATION_RESULT

24.2.4 Collecting Diagnostic Data of the Online Archival and Purge Operations

The Real-Time Purge and Archival operation via the automated scheduled task runs silently at a predefined periodicity and is non-interactive. However, you can capture and communicate the various metrics related to the purge operation, such as:

- Names of the Entity modules that were picked
- Success/failure status
- Exceptions encountered during the run
- Number of rows targeted for deletion
- Actual number of rows purged

At a minimum, these metrics are logged for every run. At any point in time, data of the most recent 500 runs is available.

The following diagnostic logging tables are part of the Real-Time Purge and Archival operation to store the diagnostic information of the entity purge runs:

- **OIM_DATAPURGE_TASK_LOG:** Stores the critical information related to the purge runs controlled by the scheduled task for the deletion of Entity data.

  Table 24–3 lists the columns of the OIM_DATAPURGE_TASK_LOG table.

Table 24–3 Columns of the OIM_DATAPURGE_TASK_LOG Table

Column	Description
OIM_DATAPRGTASK_KEY	Stores keys to uniquely identify tasks
OIM_DATAPRG_ID	Stores unique purge name
SCH_JOB_ID	Stores the Job ID of the scheduled task as assigned by the Scheduler
EXECUTION_MODE	The execution mode of the purge run, which is SCH for scheduled task mode.
PURGERUN_START_TIME	Stores the start time of the entire purge run
PURGERUN_END_TIME	Stores the end time of the entire purge run

Table 24–3 (Cont.) Columns of the OIM_DATAPURGE_TASK_LOG Table

Column	Description
PURGERUN_STATUS	Stores the overall status of the purge run, which can be any one of the following during the run: ■ STARTED ■ COMPLETED ■ ERRORED_OUT Task-level purge run could not proceed due to run-time errors. The root cause can be further probed into via the PURGE_RUN_NOTE column that stores the exception stack trace. ■ COMPLETED WITH ERROR Task-level purge run has completed but one of its modules could not get completed within the allotted time or encountered some run-time errors. The root cause can be further probed into via the PURGE_RUN_NOTE column that stores the exception stack trace.
PURGE_RUN_NOTE	Stores the task-level exception details at the purge run

- **OIM_DATAPRG_TASKS_LOGDTLS:** Stores the critical information related to the Module or Entity-level purge runs controlled by the scheduled task.

  Table 24–4 lists the columns of the OIM_DATAPRG_TASKS_LOGDTLS table.

Table 24–4 Columns of the OIM_DATAPRG_TASKS_LOGDTLS Table

Column	Description
OIM_DATPRGLOGDET_KEY	Stores keys to uniquely identify a module in a task
OIM_DATAPRGTASK_KEY	Stores the logical foreign key for the OIM_ENTITYPURGE_TASK_LOG table
MOD_NAME	Stores the module name, such as: ■ RECON ■ REQUEST ■ ORCH ■ PROVTASKS
EST_ALLOCT_TIME	Stores the time allocated for the module purge run

Table 24–4 (Cont.) Columns of the OIM_DATAPRG_TASKS_LOGDTLS Table

Column	Description
MOD_STATUS	Stores the module status, which can be any one of the following during the run: ■ STARTED ■ COMPLETED ■ COMPLETED WITH ERROR Module or Entity purge run has completed within the allotted time duration but encountered errors during its execution. The root cause can be further probed into via the MOD_PURGE_RUN_NOTE column that stores the exception stack trace. ■ ERRORED_OUT Module or Entity purge run could not proceed because of run-time errors. The root cause can be further probed into via the MOD_PURGE_RUN_NOTE column that stores the exception stack trace. ■ PARTIALLY COMPLETED Module or Entity purge run is unable to complete within the allotted time duration. This is an acceptable functional state of completion. The root cause can be further probed into via the MOD_PURGE_RUN_NOTE column that stores the exception stack trace. ■ PARTIALLY_COMPLETED WITH ERROR Module or Entity purge run could not complete within the allotted time duration but also encountered errors during its execution. The root cause can be further probed into via the MOD_PURGE_RUN_NOTE column that stores the exception stack trace.
MODPURGERUN_START_TIME	Stores the start time of the module purge run
MODPURGERUN_END_TIME	Stores the end time of the module purge run
EST_PURGE_ROW_CNT	Stores the driving table target row count for purge run for the module
ACTUAL_PURGE_ROW_CNT	Stores the actual driving table rows deleted during purge run
MOD_PURGE_RUN_NOTE	Stores the exception or other information encountered at module level

- **OIM_DATAPRG_FAILED_KEYS:** Stores the entity keys for each Module or Entity that have failed during the scheduled purge run.

  Table 24–5 lists the columns of the OIM_DATAPRG_FAILED_KEYS table.

Table 24–5 Columns of the OIM_DATAPRG_FAILED_KEYS Table

Column	Description
OIM_DATAPRGFAILED_KEY	Stores keys to uniquely identify a failed task
OIM_DATAPRGTASK_KEY	Stores the logical foreign key for the OIM_ENTITYPURGE_TASK_LOG table
MOD_NAME	Stores the module name for which the purge run fails

Table 24–5 (Cont.) Columns of the OIM_DATAPRG_FAILED_KEYS Table

Column	Description
MOD_ENTITY_KEY	Stores the driving table key for each module
ERROR_NOTE	Stores the exception stack trace

The OIM_DATAPURGE_TASK_LOG and OIM_DATAPRG_TASKS_LOGDTLS tables contain the data of the last 500 runs. The OIM_DATAPRG_FAILED_KEYS table stores the failed keys data for the last run only.

24.3 Using Command-Line Option of the Archival Purge Utilities in Oracle Identity Manager

This section describes how to use the command-line archival purge utilities. It contains the following topics:

- Understanding Command-Line Utilities
- Using the Reconciliation Archival Utility
- Using the Task Archival Utility
- Using the Requests Archival Utility

Note: You can use the Reconciliation Archival utility, the Task Archival utility, and the Requests Archival utility in both offline and online modes.

24.3.1 Understanding Command-Line Utilities

Oracle Identity Manager provides archival and purge of entity data via command-line utilities option for three entities, namely Reconciliation, Provisioning Tasks, and Requests. All the command-line utilities are part of Oracle Identity Manager installation and are interactive to capture user-specified parameters to archive and purge entity data. These utilities are available for both Linux and Microsoft Windows operating system environments.

24.3.2 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
- Troubleshooting Scenario

24.3.2.1 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 24–6 lists the active reconciliation tables with the corresponding archive reconciliation tables in which data from the active reconciliation tables are archived.

Table 24–6 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_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 horizontal tables mentioned under RECON_TABLES	"ARCH_ "first 25 characters of the horizontal tables (RA_* tables)

Note: Data from RECON_EXCEPTION table will not be archived and purged. This is due to Oracle Identity Manager predefined BIP Report dependency.

The Reconciliation Archival utility performs the following tasks:

- Archives all or specific data from the active reconciliation tables to the archive reconciliation tables
- Deletes all data from the active reconciliation tables

The Reconciliation Archival Utility archives data by moving it from the active reconciliation tables to the archive reconciliation tables based on the following two-fold criteria per the user inputs:

- The date- based criteria, which is the reconciliation event creation date. This must be specified in the YYYYMMDD format. All records on or before this date are archived.
- The functional reconciliation event state-based criteria, which is the reconciliation event status. This must be selected from the prompted status options when the utility is run.

For information about the archiving criteria, refer to "Archival Criteria" on page 14.

If you choose to archive selective data, then the utility archives reconciliation data based on selected event status that have been created on or before the specified date 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 that have been created on or before the specified date.

The files that constitute the Oracle Database version of the Reconciliation Archival utility are located in the following directory:

```
OIM_ORACLE_HOME/server/db/oim/oracle/Utilities/Recon11gArchival
```

You can run the Reconciliation Archival utility in offline mode with Oracle Identity Manager stopped, or in online mode with Oracle Identity Manager running.

Before running the utility in offline mode, you must stop Oracle Identity Manager.

24.3.2.2 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 as a DBA privilege user, for instance SYS or SYSTEM.

```
CREATE TABLESPACE OIM_RECON_ARCH
        LOGGING DATAFILE 'ORADATA/OIM_RECON_ARCH.dbf'
        SIZE 500M REUSE AUTOEXTEND ON NEXT 10M;
```

Note:

- You must replace *ORADATA* in the preceding sample command with the full path to your *ORADATA* directory.
- 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.

If you are using ASM, Exadata (ASM) or Oracle Managed Files (OMF), then follow the instructions described here.

If you are using ASM, then you can use the name of a diskgroup say `DATA 1` to create the tablespace in the database as follows:

```
CREATE TABLESPACE OIM_RECON_ARCH
       LOGGING DATAFILE '+DATA1'
       SIZE 500M AUTOEXTEND ON NEXT 10M;
```

If you are using Oracle Managed Files, then you can omit the datafile and run the command as follows:

```
CREATE TABLESPACE OIM_RECON_ARCH
       LOGGING DATAFILE
       SIZE 500M AUTOEXTEND ON NEXT 10M;
```

If you want to run the utility in offline mode, then you must stop Oracle Identity Manager before running the utility.

24.3.2.3 Archival Criteria

To select reconciliation data to archive, provide the following criteria. Data with matching values and having no reference in RECON_EXCEPTION table are archived and purged.

- Date must be in the format YYYYMMDD. All records on or before this date that match the specified reconciliation event parameter value are archived.
- Select *Closed*, *Linked*, or *Closed and Linked* for the reconciliation event parameter.
  - Closed describes events that have been manually closed in Reconciliation Manager, that is, any recon events with status as Event Closed.
  - Linked describes events that were reconciled in Oracle Identity Manager, including the following states:
    * Creation Succeeded
    * Update Succeeded
    * Delete Succeeded
  - Closed or Linked
  - Select status for reconciliation events to be archived.
    * Enter 1 for Closed
    * Enter 2 for Linked
    * Enter 3 for Closed and Linked
    * Enter 4 for Exit

24.3.2.4 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.

   Note: Oracle recommends that you run the Reconciliation Archival utility during off-peak hours.

2. If you want to run the utility in offline mode, then stop Oracle Identity Manager.

To run the utility in online mode, ignore this step and proceed to step 3.

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 to run the utility.

   On Microsoft Windows platforms, run the *path*\oim_recon_archival.bat file to run the utility.

6. For Oracle Database installations, enter values for the following parameters when prompted:

   - Oracle home directory
   - Oracle Identity Manager database user name and password

7. Enter the reconciliation creation date in the YYYYMMDD format. All records on or before this date with required status value are archived.

8. 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 Exit

9. When prompted to specify the mode of running the utility, enter 1 if you want to run the utility in online mode. Otherwise, enter 2 to run the utility in offline mode.

10. Enter the batch size for processing.

    The default batch size is 5000.

Note: Batch size is a value for the number of records to be processed in a single iteration of archival/purge, also as an internal commit at the database level. You must provide the batch size as an input parameter value while starting the operation of Archival Utilities at run time.

This batch size by default is 5000. When purging greater than few hundred thousand recon_events, a higher batch size can be opted for. This may need more resources from RDBMS, such as more space from the TEMP and UNDO tablespaces.

The utility archives the reconciliation data and provides an execution summary in a log file.

11. 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.
12. 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.

24.3.2.5 Log File Generated by the Reconciliation Archival Utility

After running the Reconciliation Archival utility, if you faces the error ORA-01034: ORACLE not available or ORA-27101: shared memory realm does not exist, then check whether the target instance is up and running.

If not then contact DBA, and bring up the instance. Ensure that target instance is accessible with OIM DB user credentials using SQLPLUS command.

24.3.2.6 Troubleshooting Scenario

While running the Reconciliation Archival utility, if the following error is encountered:

```
ORA-01034: ORACLE not available, ORA-27101: shared memory realm does not exist
```

Then, verify whether the target instance is up and running. If not, then contact the database administrator, and bring up the instance. Ensure that the target instance is accessible with Oracle Identity Manager database user credentials by using the `SQLPLUS` command.

24.3.3 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

24.3.3.1 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 **active task tables**.

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 **archive task tables**, which have the same structure as the active task tables.

Table 24–7 lists the active task tables with the corresponding archive task tables in which data from the active task tables are archived.

Table 24–7 Active and Archive Task Tables

Active Task Tables	Archive Task Tables
OSI	ARCH_OSI
OSH	ARCH_OSH
SCH	ARCH_SCH

You can use the Task Archival utility to archive the following types of tasks:

- Provisioning tasks that have been completed
- Provisioning tasks that have been completed and canceled

The Task Archival Utility archives provisioning tasks by moving it from the active task tables to the archive task tables. This is based on the following two-fold criteria per the user inputs provided:

- The date-based criteria, which is the provisioning task creation date. This must be specified in the YYYYMMDD format. All records on or before this date are archived.
- The functional criteria task status, which is the provisioning task status, for example, provisioning tasks with Completed or Completed and Canceled status. This must be selected from the prompted status options when the utility is run.

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, are 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.

You can run the Task Archival utility in offline mode with Oracle Identity Manager stopped, or in online mode with Oracle Identity Manager running.

Before running the utility in offline mode, you must stop Oracle Identity Manager.

24.3.3.2 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.

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.

24.3.3.3 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 but it is not open to other Oracle Identity Manager transactions.

Note: Oracle recommends that you run the Task Archival utility during off-peak hours.

2. Ensure that you have created a backup of the OSI, SCH, and OSH tables.
3. If you want to run the utility in offline mode, then stop Oracle Identity Manager.

To run the utility in online mode, ignore this step and proceed to step 4.

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
   - Oracle Identity Manager database user name and password
7. When prompted, select one of the following options:
   - Archive Provisioning Tasks which have been Completed.
   - Archive Provisioning Tasks which have been Completed and Cancelled.
   - Exit.
8. When prompted to specify the mode of running the utility, enter 1 if you want to run the utility in online mode. Otherwise, enter 2 to run the utility in offline mode.
9. Enter a task execution date in the format YYYYMMDD when prompted. All executed tasks, up to the task execution date you specify, are archived. To archive all tasks that were executed on or before the current date, press **Enter** without entering a date.
10. Summary information is displayed before the utility starts the archival process. The summary information gives you the total number of tasks to be archived. Read the summary information carefully and make sure your database can support the delete volume listed in the summary.

    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.

24.3.3.4 Reviewing the Output Files Generated by the Task Archival Utility

Table 24–8 describes the output files that are generated by the Task Archival utility.

Table 24–8 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.

24.3.4 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

24.3.4.1 Understanding the Requests Archival Utility

By default, Oracle Identity Manager does not remove closed or withdrawn requests from the active request tables. 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 selecting the **1 for Completed** option.
- Failed requests such as requests with status Failed, and Partially Failed. This is specified by selecting the **2 for Failed** option.

- Completed and failed requests, such as requests with status Withdrawn, Closed, Completed, Failed, and Partially Failed. This is specified by selecting the **3 for Completed and Failed** option.

Table 24–9 lists the names of the tables which are to be archived and the corresponding archival table names.

Table 24–9 Archival Tables

Main Table	Archival Table
REQUEST	ARCH_REQUEST
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
```

You can run the Requests Archival utility in offline mode with Oracle Identity Manager stopped, or in online mode with Oracle Identity Manager running.

Before running the utility in offline mode, you must stop Oracle Identity Manager.

24.3.4.2 Prerequisites for Running the Requests Archival Utility

If you want to run the utility in offline mode, then you must stop Oracle Identity Manager before running the 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.

24.3.4.3 Input Parameters

Table 24–10 lists the input parameters used by the Requests Archival utility:

Table 24–10 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, which is a TNS name or TNS alias.

Table 24–10 (Cont.) Input Parameters

Parameter	Description
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`.
Utility Running Mode	The mode in which you want to run the utility, online or offline. You must enter 1 for online mode, or 2 for offline mode. The utility runs faster when you run it in offline mode than online mode. However, running the utility in offline mode requires downtime. The archival operation can be speeded up by running in offline mode, but Oracle Identity Manager is not usable until the utility completes the archival operation. Therefore, make sure that Oracle Identity Manager is not running before choosing this option.

24.3.4.4 Running the Requests Archival Utility

To run the Requests Archival utility:

1. Ensure that the Oracle Identity Manager database is available.

   Note: It is recommended that you run the Requests Archival utility during off-peak hours.

2. If you want to run the utility in offline mode, then stop Oracle Identity Manager.

   To run the utility in online mode, ignore this step and proceed to step 3.

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.
   - 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 Failed, 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 the requests with status Request Withdrawn, Request Closed, Request Completed, Request Failed, or Request Partially Failed, and requests with creation date on or before the request creation date specified by the user in the format YYYYMMDD.

8. When prompted to specify the mode of running the utility, enter 1 if you want to run the utility in online mode. Otherwise, enter 2 to run the utility in offline mode.

9. Specify the batch size, when prompted.

   Note: Batch size is a value for the number of records to be processed in a single iteration of archival/purge also an internal commit at the database level. You must provide the batch size as an input parameter value while starting the operation of Archival Utilities at run time.

   This batch size by default is 2000. A higher batch size can be opted for, but this might require more resources from the database, such as more space from the TEMP and UNDO tablespaces.

   The utility archives the request data and provides an execution summary in a log file.

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 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.

24.3.4.5 Log Files Generated by the Utility

All the logs are written to the logs/ directory created in the current folder. Table 24–11 lists the log files generated by the utility.

Table 24–11 Logs Generated by the DB Archival Utility

Log File	Description
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

24.4 Using the Audit Archival and Purge Utility

Continuous business operations in the Oracle Identity Manager Database results in audit data growth which also has gradual increase in the storage consumption of the database server. Oracle Identity Manager's audit data related to provisioning feature are stored in legacy audit table called UPA and rest of data which are audited using lightweight auditing framework goes into AUDIT_EVENT table.

To keep this disk space consumption in control, you can use the Audit Archival and Purge utility. This utility controls the growth of the audit data by purging the data in a logical and consistent manner.

This section discusses tools and methodologies available to control the data growth in Lightweight Audit framework and Legacy Audit framework:

- Audit Data Growth Control Measures in Lightweight Audit Framework
- Audit Data Growth Control Measures in Legacy Audit Framework

24.4.1 Audit Data Growth Control Measures in Lightweight Audit Framework

To control the growth of audit data in lightweight audit framework, that is AUDIT_EVENT table, there are two available solutions:

1. Run the `Remove Audit Log Events` scheduled job.

   When this scheduled job is run, the system will automatically start purging all audit records that are older than the retention period configured in *Remove Audit Log Events older Than (in days)* field. By default it is 180 days. This scheduled job is enabled by default and has purge only option.

   For more information about Scheduled jobs, see "Scheduled Tasks" on page 18-5.

2. Partitioning AUDIT_EVENT table.

   Tip: This is a documented approach and you will need to use this solution based on your audit data compliance or lifecycle management requirement. This solution complements purge scheduled job.

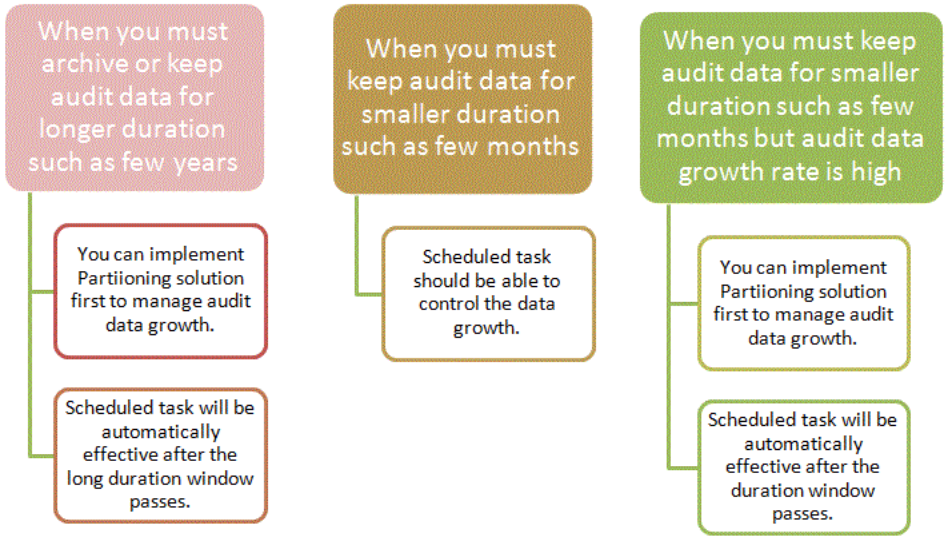
   This solution allows you to achieve the following:

   - Archiving the data, where as Schedule job allows only purging audit data.

- Flexibility to manage disk size based on your requirement. For example, data needs to be saved for a longer duration of time or if high audit growth is expected.

Figure 24–1 helps you to pick the option(s) that might be suitable to control audit data growth in your deployment.

Figure 24–1 Solutions Available to Control Audit Data Growth in Lightweight Audit Framework



This section describes how to use the Partition Based Approach to control growth of audit data in Lightweight Audit framework. It contains the following topics:

- Overview of Partition Based Approach
- Prerequisites for Partitioning the AUDIT_EVENT Table
- Preparing the AUDIT_EVENT Table for Archival and Purge
- Archiving or Purging the AUDIT_EVENT Data Using Partitions
- Ongoing Partition Maintenance

24.4.1.1 Overview of Partition Based Approach

When Partition based approach is used in combination with Scheduled job, it helps you to achieve the following solutions suitable for your deployment :

- If you need to archive or keep audit data for longer duration like, few years, then:
  1. Implement Partition based approach to manage audit data growth. This allows you to archiving and/or managing data growth.
  2. Schedule job will come into purview later when you start approaching the retention period.
- If you need to keep audit data for smaller duration, like few months, then Schedule job should be able to control the data growth.
- If you need to keep audit data for smaller duration, like few months, but audit data growth rate is high due to high number of business operations, then:

1. Implement Partition based approach to manage audit data growth.
2. Purge the data by running Schedule job.

24.4.1.2 Prerequisites for Partitioning the AUDIT_EVENT Table

The following prerequisites must be met before or when using Partition based approach:

- Licensing for Database partitioning is required to use partitioning feature of Oracle Database.
- It is recommended to try out this solution in the development or staging environment before implementing it on the production database.
- Make sure that the latest backup of the AUDIT_EVENT table is available. Creating a backup of the AUDIT_EVENT table is a compulsory prerequisite before applying this solution.
- It is recommended to use INTERVAL partitioning if your Oracle database release is 11g. Use RANGE partitioning if your oracle database is pre-11*g*. Partitioning should be done on the basis of month by using EVENT_DATE column.
- Decide how many months of audit data you require to keep online before implementing this solution. For example, if your audit data retention is six months, you have to partition AUDIT_EVENT table for six months based on month partition.
- Make sure that Oracle Identity Manager is not running and is not available for off-line utilities.You can start Oracle Identity Manager after partition for current month is created successfully and rest of audit data can be partitioned while Oracle Identity Manager is running.

24.4.1.3 Preparing the AUDIT_EVENT Table for Archival and Purge

To prepare the AUDIT_EVENT table for the audit and purge solution:

1. Query the AUDIT_EVENTS table to get the minimum and maximum calendar month for the audit data.

   Following queries can help you get the minimum and maximum month. The maximum month should be the current calendar month.

   ```
   SELECT  MIN (event_date) min_month, MAX (event_date) running_month  FROM
   AUDIT_EVENT
   ```

2. Based on the result of the previous step, three possible scenarios and time phases listed in Table 24–12 can be considered for partitioning.

Table 24–12 Possible Scenarios That are Considered For Partitioning

Scenario	Time Phase
Scenario 1	If the minimum and calendar month is the same, then you can create partition for the current month. Partitions for rest of the months are created in the future if you use INTERVAL partitioning. If RANGE partitioning is used, then you need to create future partitioning manually.
Scenario 2	If the minimum and calendar month falls within your retention duration for example six months. For example, minimum month is OCT-2015 and calendar month is DEC-2015. Then you will want to partition from OCT-2015 to DEC-2015. Future partitions are created automatically.

Table 24–12 (Cont.) Possible Scenarios That are Considered For Partitioning

Scenario	Time Phase
Scenario 3	If the minimum and calendar month falls out of your duration, like more than six months. For example, minimum month is MAY-2015 and calendar month is DEC-2015. Then, you will want to partition from JUL-2015 to DEC-2015. You will need to decide what to do with data for months (May, June) that falls out of your selected duration.

3. Refer Oracle RDBMS partitioning documentation for steps or commands to partition AUDIT_EVENT table.

24.4.1.4 Archiving or Purging the AUDIT_EVENT Data Using Partitions

Archiving or purging of audit data can be done by moving or dropping the partitions. Oracle Identify Manager does not use any partitions other than the current month. You cannot move or drop the current month partition. Which partitions to archive or purge depends on your audit data compliance or life cycle requirement.

For example if your requirement is to retain one year of audit data for compliance purpose, then follow these steps:

1. Change the retention period of audit in `Remove Audit Log Events` scheduled job from default six months (180 days) to one year.
2. Implement the partition based solution for AUDIT_EVENT table using INTERVAL or RANGE partitioning.
3. Archive or drop any partitions except the current month partition to offline storage if disk space is a concern. Oracle Identity Manager uses the current month partition to update or insert audit records. You have to keep the current month partition intact for Oracle Identity Manager to work.
4. When you are about to reach the retention duration, you may want to archive or move the partition that contains the first month data to offline storage. Otherwise, `Remove Audit Log Events` scheduled job will purge that data when it falls out of your retention period set in `Remove Audit Log Events` scheduled job.

24.4.1.5 Ongoing Partition Maintenance

- `Remove Audit Log Events` scheduled job will purging data from partitions that contains audit data older than the retention period. This creates empty partitions in AUDIT_EVENT table. It is recommended to periodically check for these empty partitions and drop them.
- Drop these empty partitions in your maintenance window using SQL like:

```
Alter table AUDIT_EVENT drop partition PARTITION_NAME UPDATE GLOBAL INDEXES;
```

24.4.2 Audit Data Growth Control Measures in Legacy Audit Framework

To control the growth in legacy audit engine, that is in UPA tables, you can use the Audit Archival and Purge utility. This utility controls the growth of the audit data by purging the data in a logical and consistent manner.

Note:

- The audit archival and purge solution is only applicable to the UPA table. It is not applicable to audit reporting tables, which are tables with the UPA_ prefix.
- The utility is compatible with Oracle Identity Manager release 9.1.0 and later.

You must shut down Oracle Identity Manager to fetch the latest data, which is to retrieve EFF_TO_DATE as null records. You can retrieve the remaining data later when Oracle Identity Manager is running with the new partitioned UPA.

Oracle recommends partitioning of the UPA table on the basis of calendar year, which allows you to archive or drop partitions. The advantage of partitioning is that the old partitions can be archived or purged because Oracle Identity Manager does not use old audit data lying in those partitions. Oracle Identity Manager uses the latest audit data and the current calendar year data. Therefore, the UPA table is partitioned based on date range-partitioning approach by calender year using EFF_TO_DATE column. After partitioning, the latest audit data where EFF_TO_DATE is NULL, can be grouped in one partition, and there is one partition for each calendar year. Oracle Identity Manager do not read or write into any other partitions except the latest and current year partitions.

For instance, if you are using Oracle Identity Manager audit feature since 2005 and implementing the audit archive and purge solution in calendar year 2011, then you will have seven partitions after this exercise, assuming that you create a partition for each calendar year. In those seven partitions, Oracle Identity Manager will only read or write the following partitions:

- The latest partition
- The partition for the current year, for example 2011

All the previous year partitions can be archived and then purged. If you do not want to archive, then you can purge those old partitions. You can reclaim the space by archiving and purging those old partitions. You must keep the latest and current year partitions untouched for Oracle Identity Manager to continue working.

This section describes how to use the Audit Archival and Purge utility. It contains the following topics:

- Prerequisites for Using the Utility
- Preparing the UPA Table for Archival and Purge
- Archiving or Purging the UPA Table

Note: The partitioning feature of Oracle Database Enterprise Edition is required for implementing audit archival and purge.

24.4.2.1 Prerequisites for Using the Utility

The following prerequisites must be met before or when using the Audit Archival and Purge utility:

- Database partitioning is supported only on Enterprise Edition of Oracle Database. Therefore, to implement the audit archival and purge solution, you must run Enterprise Edition of Oracle Database.

- The UPA table must be range-partitioned. Time interval can be any value as per data distribution. Other modes of partition methods are not supported.
- Make sure that the latest backup of the UPA table is available. Creating a backup of the UPA table is a compulsory prerequisite before applying this solution. It is recommended to try out this solution in the development or staging environment before implementing it on the production database.
- Decide how many previous year's of audit data you require to keep online before implementing this solution. This helps in creating partitions beforehand.
- Each partition should be placed on its own tablespace. Do not share the tablespace between partitions of different year or with some other data.
- During partitioning, the audit data for each calendar year is copied into a table before it is moved into a final destination. You must have provision for disk space to hold the copied data.

24.4.2.2 Preparing the UPA Table for Archival and Purge

To prepare the UPA table for the audit and purge solution:

1. Make sure that Oracle Identity Manager database has no transaction against it until the UPA table is partitioned.
2. Query the UPA table to get the minimum and maximum calendar year for the audit data. Following queries can help you get the minimum and maximum year. The maximum year should be the current calendar year.

   ```
   SELECT EXTRACT (YEAR FROM MIN (eff_to_date)) min_year,
   EXTRACT (YEAR FROM MAX (eff_to_date)) running_year FROM upa;
   ```

   This helps in deciding the partitions for each calendar year starting from minimum year.
3. Make sure that Oracle Identity Manager is not running and is not available for off-line utilities.
4. Create a new partition table.

   Assuming 2005 as minimum year and 2011 as running or current calendar year, the following decisions are to be made before creating a newly partition table:

   - How many years of old audit data you want to keep? If it is important to keep only three years of audit data, then you have to create newly partitioned table starting from year 2008. The data older than 2008 will get cleaned up when the original UPA table gets dropped.
   - After deciding the years of old data to keep, the next question is how and where the old data should be kept? Do you want to keep all the old data partitions in the active UPA table, or create backup of the old partitions and then drop the old partitions? Oracle recommends moving the old partitions into tapes and then purging them from the UPA table. As stated earlier, you must keep the latest and running calendar year partition untouched.

   The following sample assumes that you want to keep three years of audit data in UPA table and current calendar year is 2011:

   ```
   SQL> SELECT 'Create Table UPA_PART
   (
   UPA_KEY NUMBER (19) Not Null,
   USR_KEY NUMBER (19) Not Null,
   EFF_FROM_DATE TIMESTAMP (6) Not Null,
   ```

```
EFF_TO_DATE TIMESTAMP (6),
SRC VARCHAR2 (4000),
SNAPSHOT CLOB,
DELTAS CLOB,
SIGNATURE CLOB
)
PARTITION BY RANGE (EFF_TO_DATE)
(PARTITION UPA_2008 VALUES LESS THAN (TO_DATE(''01/01/2009'', ''DD/MM/YYYY''))
Tablespace upa_2008,
PARTITION UPA_2009 VALUES LESS THAN (TO_DATE(''01/01/2010'', ''DD/MM/YYYY''))
Tablespace upa_2009,
PARTITION UPA_2010 VALUES LESS THAN (TO_DATE(''01/01/2011'', ''DD/MM/YYYY''))
Tablespace upa_2010,
PARTITION UPA_2011_PART1 VALUES LESS THAN
(TO_DATE('''||TO_CHAR(SYSDATE,'DD/MM/YYYY HH24:MI:SS')||''','DD/MM/YYYY
HH24:MI:SS'')) TABLESPACE UPA_2011_PART1,
PARTITION UPA_2011_PART2 VALUES LESS THAN
(TO_DATE(''01/01/2012'',''DD/MM/YYYY'')) TABLESPACE UPA_2011_PART2,
PARTITION UPA_LATEST VALUES LESS THAN (MAXVALUE) TABLESPACE UPA_MAX
)
ENABLE ROW MOVEMENT;' FROM DUAL;
```

5. Create another non-partitioned table with similar structure as the UPA table, by running the following statement:

```
SQL> Create table upa_non_part Tablespace TBS_NAME as select * from upa where
1=2;
```

Here, *TBS_NAME* is the name of the same tablespace as of partition, which is to be exchanged.

This table is temporary in nature. The purpose of this table is to facilitate the loading of audit data to a newly partitioned UPA table.

> **Note:** UPA_NON_PART or temporary non-partitioned table must be created on same tablespace as the partition to be exchanged.

6. Load the latest audit data into the non-partitioned UPA table, as shown:

```
SQL> Insert /*+ parallel */ into upa_non_part select /*+ parallel */   * from
upa where eff_to_date is null;
SQL> COMMIT;
```

> **Note:** Using hint /*+parallel*/ in the INSERT statement is optional and you can use other hints also to improve performance according to the available resources.

7. Swap the data into the partitioned table by using the ALTER TABLE command, as shown:

```
SQL> ALTER TABLE upa_part EXCHANGE PARTITION UPA_LATEST WITH TABLE
UPA_NON_PART WITH VALIDATION UPDATE GLOBAL INDEXES;
```

8. Drop the upa_non_part table, as shown:

```
SQL> DROP TABLE upa_non_part;
```

While exchanging partitions, the data dictionary is updated instead of writing data physically. Therefore, it is necessary to drop and re-create the temporary non-partitioned UPA_NON_PART table in the same tablesapce associated to the partition to be exchanged.

9. Rename the original non-partitioned UPA table to UPA_OLD, as shown:

```
SQL> ALTER TABLE upa rename TO upa_old;
```

10. Rename the newly partitioned UPA_PART table to UPA:

```
SQL> RENAME UPA_PART to UPA;
```

11. Manage the constraints for the new UPA table. To do so:

    a. Rename the constraint from old UPA table to some other name, as shown:

```
ALTER TABLE UPA_old RENAME CONSTRAINT PK_UPA TO PK_UPA_old;
ALTER INDEX IDX_UPA_EFF_FROM_DT RENAME TO IDX_UPA_EFF_FROM_DT_old;
ALTER INDEX IDX_UPA_EFF_TO_DT RENAME TO IDX_UPA_EFF_TO_DT_old;
ALTER INDEX IDX_UPA_USR_KEY RENAME TO IDX_UPA_USR_KEY_old;
ALTER INDEX PK_UPA RENAME TO PK_UPA_OLD;
```

    b. Create the necessary indexes and primary key constraint on the newly partitioned UPA table. Make sure to add storage characteristics, such as tablespace and size. To do so, run the following SQL query:

```
SQL>create index IDX_UPA_EFF_FROM_DT on UPA (EFF_FROM_DATE) Local;
SQL>create index IDX_UPA_EFF_TO_DT on UPA (EFF_TO_DATE) Local;
SQL>create index IDX_UPA_USR_KEY on UPA (USR_KEY) Local;
SQL>ALTER TABLE UPA add constraint PK_UPA primary key (UPA_KEY) using
index;
```

Note: The global non-partitioned index is created to support the primary key. Global index becomes unusable every time a partition is touched. You must rebuild the index when required.

12. Run the statistics collection for the UPA table, as shown:

```
SQL>Exec dbms_stats.gather_table_stats(ownname => 'SCHEMA_NAME',tabname =>
'UPA',cascade => TRUE,granularity => 'GLOBAL and PARTITION');
```

Note: Global statistics must be gathered by default. Oracle 11*g* includes improvements to statistics collection for partitioned objects so untouched partitions are not rescanned. This significantly increases the speed of statistics collection on large tables where some of the partitions contain static data. When a new partition is added to the table, you need to collect statistics only for the new partition. The global statistics is automatically updated by aggregating the new partition synopsis with the existing partitions synopsis.

13. Start Oracle Identity Manager. The database is ready to be opened for transactions. Test and make sure that applications are running as expected.

14. Bring current year data in UPA_2011_PART1 to have all data and maintain consistency for current year. To do so, run the following SQL queries in sequence:

```
SQL> CREATE TABLE upa_non_part Tablespace TBS_NAME AS SELECT * FROM upa WHERE
1=2;
```

Here, *TBS_NAME* is the same tablespace name as of the partition, which is to be exchanged.

```
SQL> Alter Table UPA_NON_PART add constraint PK_UPA_NON_PART primary key
(UPA_KEY) using index;

.............
.............
SQL> Insert into upa_non_part select * from upa_old where eff_to_date >=
to_date('01/01/2011', 'mm/dd/yyyy');

.............
.............
SQL> COMMIT;

.............
.............

SQL> ALTER TABLE upa exchange partition UPA_2011_PART1 WITH table upa_non_part
WITH VALIDATION UPDATE GLOBAL INDEXES;

.............
.............
SQL> Drop table upa_non_part;
```

15. If required, bring previous year's data into the newly partitioned UPA table. To do so:

    a. Run the following SQL queries in sequence:

    ```
    SQL> CREATE TABLE upa_non_part Tablespace TBS_NAME AS SELECT * FROM upa
    WHERE 1=2;
    ```

    Here, TBS_NAME is the same tablespace as of the partition, which is to be exchanged.

    ```
    .............
    .............
    SQL> Alter Table UPA_NON_PART add constraint PK_UPA_NON_PART primary key
    (UPA_KEY) using index;
    .............
    .............
    SQL> Insert into upa_non_part select * from upa_old where eff_to_date >=
    to_date('01/01/YEAR', 'mm/dd/yyyy') and eff_to_date <
    to_date('01/01/<YEAR+1>', 'mm/dd/yyyy');
    ```

    Here, *YEAR* is the year for which you want to bring the data into newly partitioned UPA table.

    ```
    .............
    .............
    SQL>COMMIT;

    .............
    .............
    SQL> Alter table upa exchange partition UPA_<year> with table upa_non_part
    with validation Update global indexes;
    ```

b. Rebuild indexes if they are unusable. The Following SQL query shows the indexes that are unusable:

```
SQL> Select index_name, partition_name, tablespace_name, status from
user_ind_partitions;
```

c. Drop the table upa_non_part, as shown:

```
SQL> Drop table upa_non_part;
```

Note: Repeat step 15 for each old year.

16. All partition operations against UPA table are done and all the data is brought into. Run the statistics collection for the UPA table, as shown:

```
SQL>Exec dbms_stats.gather_table_stats(ownname => '<Schem_name>',tabname =>
'UPA',cascade => TRUE,granularity => 'GLOBAL and PARTITION');
```

17. Drop the UPA_OLD table if it is not required. You can create a backup of this table before dropping.

24.4.2.3 Archiving or Purging the UPA Table

Archiving and purging the UPA table is described in the following sections:

- Partitions That Must Not Be Archived or Purged
- Ongoing Partition Maintenance
- Archiving or Purging Partitions in the UPA Table

24.4.2.3.1 Partitions That Must Not Be Archived or Purged Oracle Identity Manager always requires the latest and the current calendar year audit data. The following are the names of latest and calendar year partitions:

- **UPA_LATEST:** The latest partition
- **UPA_2011_PART1** and **UPA_2011_PART2:** Partitions for the current year if current year is 2011

You must keep these two partitions untouched for Oracle Identity Manager to continue working. These two partitions should never be archived or purged.

24.4.2.3.2 Ongoing Partition Maintenance A new partition must be added to the UPA table before the new calendar year arrives. To do so, use the following SQL template:

```
SQL> Alter table UPA split partition UPA_LATEST at
(TO_DATE('01/01/YEAR+1','DD/MM/YYYY')) into (partition UPA_YEAR tablespace
UPA_YEAR,partition UPA_LATEST tablespace UPA_MAX) update global indexes;
```

Here, *YEAR* in the TO_DATE function represents the new calendar year plus one. *YEAR* for partition name and tablespace name represents new upcoming calendar year.

An example of SQL statement for adding new partition for new calendar year 2012 is as follows:

```
SQL> Alter table UPA split partition UPA_LATEST at
(TO_DATE('01/01/2013','DD/MM/YYYY')) into (partition UPA_2012 tablespace
UPA_2012,partition UPA_LATEST tablespace UPA_MAX) update global indexes;
```

Oracle recommends adding new partition with the given SQL template before the new calendar year arrives. However, if you do not add the same before the arrival of the next calender year, then the same can be done after the next year has started by using the same SQL command.

24.4.2.3.3 Archiving or Purging Partitions in the UPA Table To archive or purge partitions in the UPA table:

1. If you use the attestation feature of Oracle Identity Manager, then make sure that the partition to be archived or purged does not have any active attestation records. You can use the following SQL to verify that.

   ```
   SQL> SELECT COUNT(1) FROM UPA PARTITION(<PARTITION_TO_BE_DROPPED>)
   WHERE UPA_KEY IN (select distinct (upa_key) from apt apt, atr atr, atd atd
   where apt.atr_key=atr.atr_key and atr.atr_completion_time is NULL and
   apt.apt_key = atd.apt_key);
   ```

   This query should return zero records, which means there are no active attestation records. If this returns non-zero value, then it means that there are still active attestations pointing to the partition to be dropped. This is not common, but you must make sure that there are no active attestation records before dropping an old year partition.

2. Make sure that there are no custom reports or queries that needs the data from partition to be dropped.

3. Archive the partition to be dropped to tape or any other media. There are many ways to archive a partition. One of the ways is to use data pump or export utility to archive the partition to be dropped. Choose a way that works best in your environment.

4. Purge the partition. To do so:

   ```
   SQL> Alter table UPA drop partition PARTITION_NAME UPDATE GLOBAL INDEXES;
   SQL>Drop tablespace TBS_NAME including contents and datafiles;
   ```

   Here, TBS_NAME is the tablespace associated with the partition to be dropped, and it must not contain any other data.

   Note:

   - The current year contains two partitions named UPA_2011_PART1 and UPA_2011_PART2. When current year becomes an old year and the data for that is ready to be archived or purged, make sure to archive or purge these two partitions.
   - It is your responsibility to restore the archived data later, if required.

24.5 Using the Real-Time Certification Purge in Oracle Identity Manager

The following sections describe real-time certification purge solutions in Oracle Identity Manager:

- Understanding Real-Time Certification Purge Job
- Configuring Real-Time Certification Purge Job

Note: Real-time certification archival and purge solution is available only if you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170418. For instructions on downloading and applying the 11.1.2.3.161018 bundle patch, refer to the bundle patch documentation.

24.5.1 Understanding Real-Time Certification Purge Job

The Real-Time Certification Purge Job capability is provided by default in Oracle Identity Manager. Certification data can be continuously purged using this feature based on the options or choices made during configuration. This configuration is a one time process and the purge solution works automatically without any intervention from the administrator.

The Real-Time Certification Purge Job has the following features:

- The administrator provides values for some critical parameters by using the Scheduled Tasks section of Oracle Identity System Administration.
- Diagnostic information about each purge run is captured as a log.
- Purge tasks run periodically according to the allotted time duration.
- Data growth and subsequent footprint is controlled on an on-going basis.
- It operates online with no disruption of service.
- The purge operation via an automated scheduled task runs silently at a predefined periodicity and is non-interactive.
- Various metrics related to the purge operation, such as names of the entity modules, success or failure status, and number of rows targeted for deletion, are logged.
- These logs are diagnostic pointers for the purge operation for every run.
- Certification Purge task utilizes the existing Purge diagnostic logging framework. Refer to section Collecting Diagnostic Data of the Online Archival and Purge Operations for more information on the framework.

Oracle Identity Manager stores certification data in Oracle Identity Manager tables called active certification tables.

Naming convention used in Oracle Identity Manager in storing active certification data in the database has acronym as listed in Table 24–13:

Table 24–13 Acronyms Used in Archive Certification Tables

Table Acronym	Description
CERT_*	Table stores the certifications. CERT_ID is the key to each of these tables
CERTD_*	Table stores the decision-data for certifications
CERTDS_*	Table stores the decision-data and the snapshot-data for certifications
CERTS_*	Table stores the snapshot-data for certifications

You can use the Certification Purge Job to archive data in the archive certification tables, which have the same structure as the active certification tables.

lists the active certification tables with the corresponding archive certification tables in which data from the active certification tables are archived.

Table 24–14 Active and Archive Certification Tables

Active Certification Tables (Oracle Identity Manager Tables)	Archive Certification Tables
CERT_CERTS	ARCH_CERT_CERTS
CERT_CONFIG	ARCH_CERT_CONFIG
CERT_LAST_DECISION	ARCH_CERT_LAST_DECISION
CERT_TASK_INFO	ARCH_CERT_TASK_INFO
CERT_TASK_ACTION	ARCH_CERT_TASK_ACTION
CERT_ACTION_HISTORY_SCOPE	ARCH_CERT_ACTION_HISTORY_SCOPE
CERT_ACTION_HISTORY	ARCH_CERT_ACTION_HISTORY
CERTD_USER	ARCH_CERTD_USER
CERTD_USER_ACCT	ARCH_CERTD_USER_ACCT
CERTD_ROLE	ARCH_CERTD_ROLE
CERTD_APP_INST	ARCH_CERTD_APP_INST
CERTD_ENT_DEFN	ARCH_CERTD_ENT_DEFN
CERTD_ACCT_ENT_ASGN	ARCH_CERTD_ACCT_ENT_ASGN
CERTD_ROLE_POLICY	ARCH_CERTD_ROLE_POLICY
CERTD_POL_ENT_DEFN	ARCH_CERTD_POL_ENT_DEFN
CERTDS_USER_ROLE_ASGN	ARCH_CERTDS_USER_ROLE_ASGN
CERTDS_ENT_ASGN	ARCH_CERTDS_ENT_ASGN
CERTS_USER	ARCH_CERTS_USER
CERTS_USR_UDF	ARCH_CERTS_USR_UDF
CERTS_ROLE	ARCH_CERTS_ROLE
CERTS_APP_INST	ARCH_CERTS_APP_INST
CERTS_ENT_DEFN	ARCH_CERTS_ENT_DEFN
CERTS_ACCOUNT	ARCH_CERTS_ACCOUNT
CERTS_ACCT_ENT_ASGN	ARCH_CERTS_ACCT_ENT_ASGN
CERTS_POLICY	ARCH_CERTS_POLICY
CERTS_POL_ENT_DEFN	ARCH_CERTS_POL_ENT_DEFN
CERTS_CATALOG_UDF	ARCH_CERTS_CATALOG_UDF

Note: Certification purge is available only in online mode and via the scheduled job interface. Data from CERTD_STATS, CERT_DEFN, CERT_EVT_LSNR and CERT_EVT_TRIG tables will not be archived and purged.

For information on collecting diagnostic data of real-time certification purge job, see Collecting Diagnostic Data of the Online Archival and Purge Operations.

24.5.2 Configuring Real-Time Certification Purge Job

Certification entity data via the purge solution is continuously purged based on the selections made during configuration of the utility. You can modify these options based on data retention policies and maintenance requirements.

To configure Real-Time Certification Purge:

1. Login to Oracle Identity System Administration.
2. In the left pane, under System Configuration, click **Scheduler**.
3. Search for the **OIM Certification Purge Job**.

> **Note:** OIM Certification Purge Job is available in Oracle Identity Manager only if you have applied Oracle Identity Manager Bundle Patch 11.1.2.3.170418. For instructions on downloading and applying the 11.1.2.3.161018 bundle patch, refer to the bundle patch documentation.

4. In the Parameters section, specify values for the parameters, as described in Table 24–15:

Table 24–15 Options in the Parameters Section

Option	Description
Cert Campaigns for Purge	The purge operation runs in batches based on the input passed to this parameter. It represents the maximum number of certification campaigns to delete before a commit is issued. Default value is 10. In this field, a minimum value of 10 and a maximum value of 25 can be used.
Maximum Purge Run Duration (in Mins)	This is the maximum run duration in minutes for purge processing. Default value is 30.
Purge Criteria	This is the purge criteria and it takes the following values: ■ 1– Completed certification campaigns ■ 2– Expired certification campaigns ■ 3– Both completed certification campaigns and expired certification campaigns Default value is 1.
Purge Retention Period (in days)	This indicates the retention period in days for Certification Campaigns. Default value is 180.

> **Note:** By default, the OIM Certification Purge Job is available in the enabled state with a retention period of 180 days. You must revisit the job parameters to disable or to change the purge interval as required.

5. Click **Apply**.

Apart from the steps on the Scheduled Task UI for configuration inputs, documented in this section, there are no further manual steps required to be performed.

Note:

- For Certification Real-Time Purge operation via Scheduled Task interface, Retention Period must not be specified as ZERO as this can cause inconsistencies in purge operation.
- Simultaneously running multiple instances of the OIM Data Purge Job and the OIM Certification Purge Job is not supported via instantiation of the Scheduled Task functionality.

Part IX

Lifecycle Management

This part describes a number of additional features for Oracle Identity Manager administrators.

It contains the following chapters:

- Chapter 25, "Handling Lifecycle Management Changes"
- Chapter 26, "Securing a Deployment"

25

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

Note: In this section there are several command examples which has, password in the command, this needs to be replaced with the actual password before executing the commands.

25.1 URL Changes Related to Oracle Identity Manager

Oracle Identity Manager uses various hostnames and ports in its configuration. This section describes ways to make the corresponding changes in Oracle Identity Manager and Oracle WebLogic configuration.

This section contains the following topics:

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

25.1.1 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
- Changing Task Details URL in Human Task 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.
- When Oracle Identity Manager managed server is enabled for SSL port, perform the procedures described in these sections to change the Oracle Identity Manager port and protocol, such as t3 to t3s and http to https.

25.1.1.1 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 a load balancer or web server or a 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**.

   In a clustered deployment, when you select **oracle.iam** under Application Defined MBeans, Oracle Identity Manager server name is displayed. Select the server and continue with the navigation.

   Note: In a clustered deployment, the change to the OimFrontEndURL must be made on each server in the cluster.

5. Enter new value for the OimFrontEndURL attribute, and click **Apply** to save the changes. Example values can be:

   http://*OIM_SERVER*:*OIM_PORT*

   https://server1.mycompany.com

   https://server1.mycompany.com:14002

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.

25.1.1.2 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://server1.mycompany.com:8001

   t3://server1.mycompany.com:8001,server2.mycompany.com:9001

Note:

- The value of the BackOfficeURL attribute must be empty for Oracle Identity Manager nonclustered and clustered deployments.
- For SSL-enabled Oracle Identity Manager setup, BackOfficeURL attribute must be populated with the correct URL, for example:

  t3s://*OIM_HOST*:*OIM_SSL_PORT*

25.1.1.3 Changing Task Details URL in Human Task Configuration

The task details URL is the URL to display the task details page for a particular human task in Inbox. This can be a load balancer URL or Web server URL depending on whether the application server is fronted with load balancer, or Web server, or single application server URL.

The change might be required 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 task details URL in human task configuration:

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

   http://*ADMIN_SERVER*/em

   For a clustered deployment, ensure that at least one SOA server in the SOA cluster is running.
2. Navigate to **SOA**, **soa-infra(*SOA_SERVER_NAME*)**, **default**.
3. Click **DefaultRequestApproval**.
4. In the Component Metrics section, click the **ApprovalTask** link.
5. Click the **Administration** tab.
6. Make the required changes to Host Name, HTTP Port, and HTTPS Port.
7. Repeat steps 5 and 6 for all other human tasks in DefaultRequestApproval, for example ChallengeTask.
8. Repeat steps 4 to 7 for all other composites.

25.1.2 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.
- When Oracle Identity Manager database is enabled for SSL port, perform this procedure to change the Oracle Identity Manager database URL and properties accordingly.

- **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 soaOIMLookupDB configuration:**
  1. Navigate to **Services**, **JDBC**, **Data Sources**, and then **soaOIMLookupDB**.
  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 **oimOperationsDB**.
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 ApplicationDB configuration:**
  1. Navigate to **Services**, **JDBC**, **Data Sources**, and then **ApplicationDB**.
  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:**

  Note: This step is required only if database host and port of MDS schema is changed.

  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.
     - For SSL-enabled database, the changes described in this section are not applicable.

       For DB changes related to SSL, follow the instructions provided in "Updating Oracle Identity Manager Authenticators" on page 25-37.

After making changes in the datasources, restart the Oracle WebLogic Administrative Server, and start the Oracle Identity Manager managed WebLogic servers.

Note: Whenever Oracle Identity Manager application configuration information is to be changed by using OIM App Config MBeans from the Enterprise Management (EM) console, at least one of the Oracle Identity Manager Managed Servers must be running. Otherwise, you cannot figure out any of the OIM App Config MBeans from the EM console.

- **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. For information about these generic changes, see *Oracle Fusion Middleware High Availability Guide*.

     See "Oracle Identity Manager Database Host and Port Changes" on page 25-4 for information about changing the port at the database.

- **To change the Oracle Identity Manager database host and port in BI Publisher:**

  1. Login to BI Publisher.
  2. Click the **Administration** tab.
  3. Click **JDBC Connection** under Data Sources.
  4. Click **OIM JDBC**, and change the database host and port.
  5. Click **Test Connection**. The connection is established successfully after confirmation.
  6. Click **Apply**.

- **Perform the following additional steps if Oracle Identity Manager is made to point to another database of another Oracle Identity Manager instance instead of current database port being changed:**

  1. Copy .xldatabasekey from Oracle Identity Manager that is installed on the destination DB to the source Oracle Identity Manager deployment. Copy

DOMAIN_HOME/config/fmwconfig/.xldatabasekey from destination to source Oracle Identity Manager.

2. Copy the following keys from Oracle Identity Manager deployment on the destination DB to the source deployment:

   OIMSchemaPassword

   .xldatabasekey

   DataBaseKey

3. To get the Oracle Identity Manager credential store from Oracle Identity Manager installed on the destination DB:

   a. Login to Oracle Enterprise Manager by using the following URL:

      http://*HOST*:*ADMIN_SERVER_PORT*>/em

   b. Navigate to Weblogic Domain, right-click ***DOMAIN_NAME***, and select **System MBean Browser**.

   c. Under Application Defined MBeans, navigate to **com.oracle.jps**, **Server:*OIM_SERVER_NAME***, **JpsCredentialStore**.

   d. Go to **Operations**, **getPortableCredentialMap**. Enter the parameter value as `oim` and `Invoke`.

      This displays the oim credential map. Note the passwords for OIMSchemaPassword, .xldatabasekey, and DataBaseKey.

4. To change the keys in the OIM credential store on the source deployment:

   a. **OIMSchemaPassword:** Navigate to Weblogic Domain, right-click ***DOMAIN_NAME***, and navigate to **Security**, **Credentials**. Expand **oim**, and click **OIMSchemaPassword**. Click **Edit**, and enter the new password in Password and Confirm Password fields.

   b. **.xldatabasekey:** Repeat the same steps for .xldatabasekey.

   c. **DataBaseKey:** Repeat the same steps for DataBaseKey.

25.1.3 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 System Administration.
2. Under Provisioning Configuration, click **IT Resource**.
3. From the IT Resource Type list, select **Directory Server** , and click **Search**.
4. 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**.

See Also: See "Updating Oracle Identity Manager for libOVD details" on page 25-39 for information about changing OVD port at OVD/LDAP server.

25.1.4 BI Publisher Host and Port Changes

To change BI Publisher 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.DiscoveryConfig**, **Discovery**.
5. Enter a new value for the BIPublisherURL attribute, and click **Apply** to save the changes.
6. To change the BI Publisher host and port in jms_cluster_config.properties file:
   a. Go to the *DOMAIN_NAME*/config/bipublisher/repository/Admin/Scheduler/ directory.
   b. In a text editor, open the jms_cluster_config.properties file, and replace the BI Publisher host and port.
   c. Save the jms_cluster_config.properties file.
   d. Restart BI Publisher server.

25.1.5 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.
- When SOA managed server is enabled for SSL port, perform the procedure described in this section to change the SOA port and protocol, such as t3 to t3s and http to https.

---

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 value of the Rmiurl attribute, 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://soaserver1.mycompany.com:8001

   t3s://mysoaserver1.mycompany.com:8002,mysoa1.mycompany.com:8002

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

6. Change the SOA JNDIProvider host and port. To do so:
   a. Login to WebLogic Administration Console.
   b. In the Domain Structure section, navigate to ***OIM_DOMAIN***, **Services**, **Foreign JNDI Providers**.
   c. Click **ForeignJNDIProvider-SOA**.
   d. In the Configuration tab, verify that the **General** subtab is active.
   e. Change the value of Provider URL to the Rmiurl provided in Step 5.

25.1.6 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 for a clustered deployment, are running:

   http://*ADMIN_SERVER*/em

2. Navigate to **Identity and Access**, and then to **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.SSOConfig**, and then **SSOConfig**.
5. Change the values of the AccessServerHost and AccessServerPort attributes and other attributes as required, and click **Apply** to save the changes.

25.2 Password Changes Related to Oracle Identity Manager

Various passwords are used for Oracle Identity Manager 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 Manager 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 Administrator Database Password
- Changing Oracle Identity Manager Database Password
- Changing Oracle Identity Manager Passwords in the Credential Store Framework

- Changing OVD Password
- Changing Oracle Identity Manager Administrator Password in LDAP
- Unlocking Oracle Identity Manager Administrator Password in LDAP
- Changing Schema Passwords

25.2.1 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**.

Weblogic credentials must be updated in the following places:

1. Foreign JNDI Provider. To do so:
   a. Login to WebLogic Administrative Console.
   b. In the Domain Structure section, navigate to **OIM_DOMAIN**, **Services**, **Foreign JNDI Providers**.
   c. Click **ForeignJNDIProvider-SOA**.
   d. In the Configuration tab, verify that the General subtab is active.
   e. Provide weblogic user's new password in the password and confirm password fields.
2. SOAAdminPassword in CSF. See "Changing Oracle Identity Manager Passwords in the Credential Store Framework" on page 25-16 for details.

25.2.2 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 "Changing Password" in the *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Manager*.

When you change the Oracle Identity Manager system administrator password, you must also update the password in the `sysadmin` key under the `oim` map in CSF. See "Changing Oracle Identity Manager Passwords in the Credential Store Framework" on page 25-16 for information about the CSF keys.

Note: If OAM or OAAM is integrated with Oracle Identity Manager, then you must 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/technetwork/indexes/documentation/index.html

25.2.3 Changing Oracle Identity Manager Administrator Database Password

This section describes resetting Oracle Identity Manager password in the following types of deployments:

- Oracle Identity Manager deployment without LDAP synchronization
- Oracle Identity Manager deployment with LDAP synchronization enabled
- Oracle Identity Manager deployment that is integrated with Access Manager (OAM)

Resetting System Administrator password can be performed by using the oimadminpasswd_wls.sh utility, which is available in the *OIM_HOME*/server/bin/ directory. The steps to run the oimadminpasswd_wls.sh utility are the same for both types of deployment: Oracle Identity Manager with LDAP synchronization enabled and without LDAP synchronization enabled.

This section describes resetting Oracle Identity Manager password in the following topics:

- Resetting System Administrator Database Password in Oracle Identity Manager Deployment
- Resetting System Administrator Database Password When Oracle Identity Manager Deployment is Integrated With Access Manager

25.2.3.1 Resetting System Administrator Database Password in Oracle Identity Manager Deployment

To reset System Administrator database password:

1. As a prerequisite for running the oimadminpasswd_wls.sh utility, open the *OIM_HOME*/server/bin/oimadminpasswd_wls.properties file in a text editor, and set values for the following properties:
   - JAVA_HOME: Set this to jdk6 or later, for example:

     ```
     JAVA_HOME=/opt/softwares/shiphome/jdk170_131
     ```

   - COMMON_COMPONENTS_HOME: This is Oracle Middleware common home directory, for example:

     ```
     COMMON_COMPONENTS_HOME=/opt/softwares/shiphome/oracle_common
     ```

   - OIM_ORACLE_HOME: This is Oracle Identity Manager Oracle home directory, for example:

     ```
     OIM_ORACLE_HOME=/opt/softwares/shiphome/Oracle_IDM1
     ```

   - ORACLE_SECURITY_JPS_CONFIG: Specify the jps-config-jse.xml file location present in Oracle Identity Manager Domain, for example:

```
ORACLE_SECURITY_JPS_CONFIG=/opt/softwares/shiphome/user_projects/domains/b
ase_domain/config/fmwconfig/jps-config-jse.xml
```

- DOMAIN_HOME: Specify Oracle Identity Manager Domain Home location of the Weblogic Application Server, for example:

  ```
  DOMAIN_HOME=/opt/softwares/shiphome/user_projects/domains/base_domain
  ```

- DBURL: Oracle Identity Manager database URL, for example:

  ```
  DBURL=jdbc:oracle:thin:@dbhostname:5521:orclsid
  ```

- DBSCHEMAUSER: Oracle Identity Manager schema username, for example:

  ```
  DBSCHEMAUSER=DEV_OIM
  ```

- OIM_OAM_INTG_ENABLED: Set this to false if Oracle Identity Manager deployment is not integrated with Access Manager, for example:

  ```
  OIM_OAM_INTG_ENABLED=false
  ```

Note: Other properties, such as LDAPURL, LDAPADMINUSER, and OIM_ADMIN_LDAP_DN can be ignored as they are used only in an integrated setup between Oracle Identity Manager and Access Manager.

2. Go to the *OIM_HOME*/server/bin/ directory, and run the following command:

   ```
   sh oimadminpasswd_wls.sh oimadminpasswd_wls.properties
   ```

   The following is a sample output:

   ```
   Enter OIM DB Schema Password :
   Enter OIM Adminstrator xelsysadm new Password:
   Re-enter OIM Adminstrator xelsysadm new Password:
   WARNING: Not able to fetch OIMPlatform instance for the given Platform. Hence
   defaulting to the OIMWebLogicPlatform

   OIM Admin user xelsysadm password reset successfully in OIMDB
   ```

Note: The warning messages that are displayed while running the oimadminpasswd_wls.sh script can be ignored.

25.2.3.2 Resetting System Administrator Database Password When Oracle Identity Manager Deployment is Integrated With Access Manager

If Oracle Identity Manager is integrated with OAM, then LDAP directory, such as Oracle Internet Directory, is used for all authentication purposes. Therefore, Oracle Identity Manager Administrator xelsysadm password is reset in LDAP. Although the xelsysadm password present in Oracle Identity Manager database is not used in this topology, it is also reset along with LDAP directory to ensure that the passwords in both repositories are in sync.

To reset System Administrator database password when Oracle Identity Manager Deployment is Integrated With Access Manager:

1. As a prerequisite for running the oimadminpasswd_wls.sh utility, open the *OIM_HOME*/server/bin/oimadminpasswd_wls.properties file in a text editor, and set values for the following properties:

   - JAVA_HOME: Set this to jdk6 or later, for example:

     ```
     JAVA_HOME=/opt/softwares/shiphome/jdk170_131
     ```

   - COMMON_COMPONENTS_HOME: This is Oracle Middleware common home directory, for example:

     ```
     COMMON_COMPONENTS_HOME=/opt/softwares/shiphome/oracle_common
     ```

   - OIM_ORACLE_HOME: This is Oracle Identity Manager Oracle home directory, for example:

     ```
     OIM_ORACLE_HOME=/opt/softwares/shiphome/Oracle_IDM1
     ```

   - ORACLE_SECURITY_JPS_CONFIG: Specify the jps-config-jse.xml file location present in Oracle Identity Manager Domain, for example:

     ```
     ORACLE_SECURITY_JPS_CONFIG=/opt/softwares/shiphome/user_projects/domains/base_domain/config/fmwconfig/jps-config-jse.xml
     ```

   - DOMAIN_HOME: Specify Oracle Identity Manager Domain Home location of the Weblogic Application Server, for example:

     ```
     DOMAIN_HOME=/opt/softwares/shiphome/user_projects/domains/base_domain
     ```

   - DBURL: Oracle Identity Manager database URL, for example:

     ```
     DBURL=jdbc:oracle:thin:@dbhostname:5521:orclsid
     ```

   - DBSCHEMAUSER: Oracle Identity Manager schema username, for example:

     ```
     DBSCHEMAUSER=DEV_OIM
     ```

   - OIM_OAM_INTG_ENABLED: Set this to true if Oracle Identity Manager deployment is integrated with Access Manager, for example:

     ```
     OIM_OAM_INTG_ENABLED=true
     ```

   - LDAPURL: LDAP directory URL. Non-SSL port must be specified, for example:

     ```
     LDAPURL=ldap://LDAP_HOSTNAME:3060
     ```

   - LDAPADMINUSER : LDAP directory admin username, for example:

     ```
     LDAPADMINUSER=cn=orcladmin
     ```

   - OIM_ADMIN_LDAP_DN: Oracle Identity Manager Administrator xelsysadm complete DN in the LDAP directory, for example:

     ```
     OIM_ADMIN_LDAP_DN=cn=xelsysadm,cn=Users,dc=us,dc=mycompany,dc=com
     ```

2. Go to the *OIM_HOME*/server/bin/ directory, and run the following command:

   ```
   sh oimadminpasswd_wls.sh oimadminpasswd_wls.properties
   ```

   The following is a sample output:

   ```
   Enter OIM DB Schema Password :
   Enter OIM Adminstrator xelsysadm new Password:
   Re-enter OIM Adminstrator xelsysadm new Password:
   ```

```
WARNING: Not able to fetch OIMPlatform instance for the given Platform. Hence
defaulting to the OIMWebLogicPlatform

OIM Admin user xelsysadm password reset successfully in OIMDB
OIM Admin user cn=xelsysadm,cn=Users,dc=...,dc=...,dc=... password reset
successfully in LDAP
```

Note:

- The warning messages that are displayed while running the oimadminpasswd_wls.sh script can be ignored.
- The xelsysadm password expiry setting is not set to expire until 2035. During integration between Oracle Identity Manager and Access Manager, the obpasswordexpirydate setting for the xelsysadm user is set to `"2035-01-01T00:00:00Z"`. If this value has been changed, then revert it to `"2035-01-01T00:00:00Z"` for xelsysadm. This value is initially loaded from a following template LDIF file:

  $*OIM_ORACLE_HOME*/idmtools/templates/oid/idm_xelsysadmin_user.ldif

25.2.4 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 ApplicationDB configuration:**
  1. Navigate to **Services**, **JDBC**, **Data Sources**, **ApplicationDB**.
  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 soaOIMLookupDB configuration:**

1. Navigate to **Services**, **JDBC**, **Data Sources**, and then **soaOIMLookupDB**.
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.
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**.

- **To change the Oracle Identity Manager database password in BI Publisher:**
  1. Login to BI Publisher.
  2. Click the **Administration** tab.
  3. Click **JDBC Connection** under Data Sources.
  4. Click **OIM JDBC**, and change the password in the Password field.
  5. Click **Test Connection**. The connection is established successfully after confirmation.
  6. Click **Apply**.

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

25.2.5 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 25–1 lists the keys and the corresponding values:

Table 25–1 CSF Keys

Key	Description
DataBaseKey	The password for the key used to encrypt database. The password is the user input value in the installer for the Oracle Identity Manager keystore.
.xldatabasekey	The password for keystore that stores the database encryption key. The password is the user input value in the installer for the Oracle Identity Manager keystore.
xell	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	The password for the default_keystore.jks JKS keystore in the *DOMAIN_HOME*/config/fmwconfig/ directory. The password is the user input value in the installer for the Oracle Identity Manager keystore.
SOAAdminPassword	The password is user input value in the installer for SOA Administrator Password field.
OIMSchemaPassword	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	The password is the user input value in the installer for the Oracle Identity Manager keystore.

To change the values of the CSF keys:

1. Login to Oracle Enterprise Manager by navigating to the following URL:

   http://*ADMIN_SERVER*/em
2. Navigate to **Weblogic Domain**, ***DOMAIN_NAME***.
3. Right-click **oim**, and select **Security**, **Credentials**.

4. Edit the Directory Server IT resource. To do so, in the Admin Password field, enter the new OVD password, and click **Update**.

25.2.6 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**.

25.2.7 Changing Oracle Identity Manager Administrator Password in LDAP

To change Oracle Identity Manager System Administrator password in LDAP in a Oracle Identity Manager deployment that is SSO enabled and integrated with Access Manager (OAM):

1. Look up the dn for the user from LDAP, as shown:

```
$ORACLE_HOME/bin/ldapsearch -D cn=orcladmin -w fusionapps1 -h localhost -p
6501 -b dc=com "cn=SYS_ADMIN" orclaccountlocked dn
```

   Here, *SYS_ADMIN* is the System Administrator user login.

2. Create a file similar to the following:

```
$ more /tmp/resetpassword_SYS_ADMIN

dn: cn=SYS_ADMIN,cn=Users,dc=us,dc=mycompany,dc=com
changetype: modify
replace: userPassword
userPassword: NEW_PASSWORD
```

   Here, *NEW_PASSWORD* is the password that you want in clear text.

3. Change the password, as shown:

```
$ORACLE_HOME/bin/ldapmodify -D cn=orcladmin -w fusionapps1 -h localhost -p
6501 -f /tmp/ resetpassword _SYS_ADMIN
```

4. Verify that the user password is changed, as shown:

```
$ORACLE_HOME/bin/ldapbind -D 'cn=SYS_ADMIN,cn=Users,dc=us,dc=mycompany,dc=com'
-w NEW_PASSWORD -h localhost -p 6501
```

25.2.8 Unlocking Oracle Identity Manager Administrator Password in LDAP

To unlock Oracle Identity Manager System Administrator password in LDAP in a Oracle Identity Manager deployment that is SSO enabled and integrated with OAM:

1. Look up the dn for the user from LDAP, as shown:

```
$ORACLE_HOME/bin/ldapsearch -D cn=orcladmin -w fusionapps1 -h localhost -p
6501 -b dc=com "cn=SYS_ADMIN" orclaccountlocked dn
```

If orclaccountlocked has a value of 1, then it means that the user is locked.

2. Create a file similar to the following:

```
$ more /tmp/unlock_SYS_ADMIN

dn: cn=SYS_ADMIN,cn=Users,dc=us,dc=mycompany,dc=com
changetype: modify
replace: orclaccountlocked
orclaccountlocked: 0
```

3. Unlock the user, as shown:

```
$ORACLE_HOME/bin/ldapmodify -D cn=orcladmin -w fusionapps1 -h localhost -p
6501 -f /tmp/unlock_SYS_ADMIN
```

4. Verify that the user is unlocked, as shown:

```
$ORACLE_HOME/bin/ldapsearch -D cn=orcladmin -w fusionapps1 -h localhost -p
6501 -b dc=com "cn=SYS_ADMIN" orclaccountlocked dn
```

The value of orcladdountlocked must be 0.

25.2.9 Changing Schema Passwords

To change OIM, MDS, SOAINFRA, OPSS, ORASDPM, and BI Publisher schema passwords:

1. Stop all the Managed Servers and application server.
2. Create a backup of the entire domain and the database.
3. Start the application server.
4. Change the xxxx_OPSS user password. To do so:
   a. Run the following command:

```
SQL> alter user xxxx_OPSS identified by NEW_PASSWORD;
```

   b. Go to the *ORACLE_COMMON*/common/bin/ directory, and run the `wlst` command.
   c. Run the `modifyBootStrapCredential` script, as shown:

```
modifyBootStrapCredential(jpsConfigFile='DOMAIN_NAME/config/fmwconfig/jps-
config.xml', username='xxxx_OPSS', password='NEW_PASSWORD')
```

5. Login to Weblogic Administrative Console. Navigate to **Services**, **Data Sources**.
6. Select **opss-DBDS**, **Connection Pool**, and enter the new password set to xxxx_opss in step 4a. Save the changes.
7. Restart the application server, but do not start the Managed Servers.
8. Connect to the database with sqlplus as system user, and then run the following commands:
   a. To change the password for xxx_OIM, run:

```
SQL> alter user xxx_OIM identified by NEW_PASSWORD;
```

   b. To change the password for xxx_MDS, run:

```
SQL> alter user xxx_MDS identified by NEW_PASSWORD;
```

c. To change the password for xxx_SOAINFRA, run:

```
SQL> alter user xxx_SOAINFRA identified by NEW_PASSWORD;
```

d. To change the password for xxx_ORASDPM, run:

```
SQL> alter user xxx_ORASDPM identified by NEW_PASSWORD;
```

e. To change the password for xxx_BIPLATFORM, run:

```
SQL> alter user xxx_BIPLATFORM identified by NEW_PASSWORD;
```

9. Verify that the passwords have been changed. To do so, login to the database with sqlplus and the four users and the new passwords.
10. Login to the WebLogic Administrative Console.
11. Go to **Services**, **Data Sources**, and then perform the following:
    a. Select **soaOIMLookupDB**, **Connection Pool**, and enter the new password set to xxx_OIM in step 12a.
    b. Select **oimJMSStoreDS**, **Connection Pool**, and enter the new password set to xxx_OIM in step 12a.
    c. Select **oimOperationsDB**, **Connection Pool**, and enter the new password set to xxx_OIM in step 12a.
    d. Select **ApplicationDB**, **Connection Pool**, and enter the new password set to xxx_OIM in step 12a.
    e. Select **mds-oim**, **Connection Pool**, and enter the new password set to xxx_MDS in step 12b.
    f. Select **mds-owsm**, **Connection Pool**, and enter the new password set to xxx_MDS in step 12b.
    g. Select **mds-soa**, **Connection Pool**, and enter the new password set to xxx_MDS in step 12b.
    h. Select **EDNDataSource**, **Connection Pool**, and enter the new password set to xxx_SOAINFRA in step 12c.
    i. Select **EDNLocalTxDataSource**, **Connection Pool**, and enter the new password set to xxx_SOAINFRA in step 12c.
    j. Select **SOADataSource**, **Connection Pool**, and enter the new password set to xxx_SOAINFRA in step 12c.
    k. Select **SOALocalTxDataSource**, **Connection Pool**, and enter the new password set to xxx_SOAINFRA in step 12c.
    l. Select **OraSDPMDataSource**, **Connection Pool**, and enter the new password set to xxx_ORASDPM in step 12d.
12. Change OIMAuthenticationProvider configuration. To do so:
    a. In the WebLogic Administrative Console, navigate to **Security Realms**, **myrealm**, and then **Providers**.
    b. Click **OIMAuthenticationProvider**.
    c. Click **Provider Specific**.
    d. In the DBPassword field, enter the new Oracle Identity Manager database schema password.

e. Click **Save** to save the changes.

13. Change the domain credential store configuration. To do so:
    a. Login to Oracle Enterprise Manager.
    b. Navigate to **Weblogic Domain**, and then ***DOMAIN_NAME***.
    c. Right-click the domain name, and select **Security**, **Credentials**, and then **oim**.
    d. Select **OIMSchemaPassword**, and click **Edit**.
    e. In the Password field, enter the new password, and then click **OK**.
14. Change the oim and soa schema password in BI Publisher. To do so:
    a. Login to BI Publisher.
    b. Click the **Administration** tab.
    c. Click **JDBC Connection** under Data Sources.
    d. Click **OIM JDBC**, and change the password in the Password field.
    e. Click **Test Connection**. The connection is established successfully after confirmation.
    f. Click **Apply**.
    g. Repeat the steps 14d through 14f for JDBC data source `BPEL JDBC`.
15. If BI Publisher schema password is changed, then perform the following steps:
    a. Login to Oracle Enterprise Manager.
    b. Expand **WebLogic Domain**, ***DOMAIN_NAME***.
    c. Under the ***DOMAIN_NAME*** on the right pane, from the WebLogic Domain list, select **JDBC Data Sources**.
    d. Select **bip_datasource** in the table, and then click **Edit** on the toolbar.
    e. Click the **Connection Pool** tab. In the Database Connection Information section, change the password, and then click **Apply** on the upper right corner.
    f. Start BI Publisher services.
16. Restart WebLogic Admin Server.
17. Start the SOA and Oracle Identity Manager Managed Servers.

25.3 Configuring SSL for Oracle Identity Manager

This section describes the procedure for generating keys, signing and exporting certificates, setting up SSL Configuration for Oracle Identity Manager and for the components with which Oracle Identity Manager interacts, and establish secure communication between them.

A SHA-2 compliant certificate is a prerequisite for using TLS 1.2 protocol for SSL communication.

Note:

- For information related to IBM Java 7, SR4 version support of SHA-2 cipher suites and Transport Layer Security (TLS) version 1.2 refer to IBM documentation.
- In the following sections several examples are provided. They have parameters which are used to enable more debugging information and are optional. For example, `-Dweblogic.StdoutDebugEnabled=true -Dssl.debug=true -Djavax.net.debug=ssl:handshake:verbose`.

For Oracle JDK 7, download and apply latest Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files. Relocate the local_policy.jar and US_export_policy.jar jars files into <JAVA_HOME>/jre/lib/security directory.

Note: If Opatch version is lesser than 12.1.0.1.10, then upgrade the OPatch utility by applying p21142429_121010_Linux-x86-64.zip patch.

Apply p23176395_121020_Generic.zip patch to DB_HOME to get the support of TLS 1.2 on Oracle 12c DB (12.1.0.2).

Apply p19030178_111190_Generic.zip patch on oracle_common directory.

Apply p13964737_1036_Generic.zip Weblogic patch via BSU if Demo Identity and Demo trust is used at Weblogic Level.

It includes the following topics:

- Generating Custom Key Stores (Optional)
- Configuring Custom Key Stores (Optional)
- Enabling SSL for Oracle Identity Manager and SOA Servers
- Enabling SSL for Oracle Identity Manager DB
- Enabling SSL for SOA Approval Composites
- Enabling SSL for LDAP Synchronization
- Configuring SSL for Design Console
- Configuring SSL for Oracle Identity Manager Utilities with TLS

Note:

- Section "Generating Custom Key Stores (Optional)" on page 25-22 provides example commands that are used later in the document. These are for reference and not part of the mandatory steps of configuration.
- For configuring Oracle User Messaging Service (UMS) notification that is SSL-based, see "Using UMS for Notification" on page 19-2.

  For more details on configuring UMS to connect to a mail server with SSL, see "Configuring Oracle User Messaging Service" in the *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

25.3.1 Generating Custom Key Stores (Optional)

This section includes the following topics:

- Generating Keys
- Signing the Certificates
- Exporting the Certificate
- Importing the Certificate

Note: The procedures described in sections "Generating Keys" on page 25-22 to "Importing the Certificate" on page 25-23 are optional. These steps are required if you have custom identity and trust store for WebLogic servers.

SSL can be enabled with default identity and trust store as well.

25.3.1.1 Generating Keys

You can generate private and public certificate pairs by using the keytool command. The syntax is:

```
$JAVA_HOME/jre/bin/keytool -genkey -alias ALIAS -keyalg ALGORITHM -keysize
KEY_SIZE -sigalg SIGN_ALORITHM -dname DISTINGUISHED_NAME -keypass KEY_PASSWORD
-keystore KEYSTORE_NAME -storepass KEYSTORE_PASSWORD
```

The following example creates an identity keystore named oimsupporttrust.jks:

```
$JAVA_HOME/jre/bin/keytool -genkey
-alias supportpvtkey
-keyalg RSA -keysize 2048
-sigalg SHA256withRSA
-dname "CN=oimhost.mycompany.com, OU=Identity, O=Oracle Corporation,C=US"
-keypass privatepassword
-keystore oimsupportidentity.jks
-storepass password
```

When generating the certificate for Oracle Identity Manager, in CN attribute specify the host name where Oracle Identity Manager is deployed. Similarly, when generating the certificate for SOA, in CN attribute specify the host name where SOA is deployed. For example:

```
-dname "CN=myhost.us.example.com, OU=Identity, O=Example Corporation,C=US"
```

Note:

- Change the parameter values passed to the keytool command according to your requirements. Ensure that there is no line break in the keytool argument.
- The custom identity keystore, `oimsupportidentity.jks` must be created or copied under WL_HOME/server/lib/.
- If JDK 7u40 or later is used, then the value of the keysize option must be greater than or equal to 1024. For more information about this limitation, see "Default x.509 Certificates Have Longer Key Length" at the following URL:

  http://www.oracle.com/technetwork/java/javase/7u40-relnotes-2004172.html

25.3.1.2 Signing the Certificates

Use the following keytool command to sign the certificates that you created:

```
$JAVA_HOME/jre/bin/keytool -selfcert -alias supportpvtkey
-sigalg SHA256withRSA -validity 2000 -keypass <privatepassword>
-keystore oimsupportidentity.jks
-storepass <password>
```

Note: Change the parameter values passed to the keytool command according to your requirements. Ensure that there is no line break in the keytool argument.

25.3.1.3 Exporting the Certificate

Use the keytool command to export the certificate from the identity keystore to a file. The syntax is:

```
$JAVA_HOME/jre/bin/keytool -export -alias ALIAS -file FILE_TO_EXPORT -keypass
KEY_PASSWORD -keystore KEYSTORE_NAME -storepass KEYSTORE_PASSWORD
```

For example, the following example command exports the certificate to a file named supportcert.pem:

```
$JAVA_HOME/jre/bin/keytool -export -alias supportpvtkey
  -file supportpvtkeycert.pem
  -keypass <password>
  -keystore oimsupportidentity.jks
  -storepass <password>
```

25.3.1.4 Importing the Certificate

Use the keytool command to import the certificate from a file. The syntax is:

```
keytool -import -alias ALIAS -trustcacerts -file FILE_TO_IMPORT -keystore
KEYSTORE_NAME -storepass KEYSTORE_PASSWORD
```

In the following example, the certificate file supportcert.pem is imported to the identity keystore oimsupporttrust.jks with password weblogic1:

```
$JAVA_HOME/jre/bin/keytool -import -alias supportpvtkey -trustcacerts -file
supportpvtkeycert.pem
```

```
-keystore oimsupporttrust.jks -storepass <password>
```

Note:

- Change the parameter values passed to the keytool command according to your requirements. Ensure that there is no line break in the keytool argument.
- This custom trust keystore oimsupporttrust.jks must be created or copied under DOMAIN_HOME/config/fmwconfig/.
- This command loads a trusted CA certificate into a keystore. If the keystore does not exist, it is created.

25.3.2 Configuring Custom Key Stores (Optional)

Perform the following steps to configure custom key stores:

Note: See "Generating Custom Key Stores (Optional)" for information about generating custom keys.

1. In the WebLogic Server Administration Console, click **Environment**, **Servers**, **Server_Name (OIM_Server1)**, **Configuration**, and then **General**.
2. Click **Lock & Edit**.
3. Select SSL listen port enabled. The default SSL port is 14002 and 14001 for non-SSL.
4. Select the Keystores tab.
5. From the Keystore list, select **Custom Identity and Custom Trust**.

   Note: If you have created only custom identity and using java standard trust, then select the **Custom Identity**, **Java Standard Trust** option.

   If you have created custom identity and custom trust, then select the **Custom Identity and Custom Trust** option.

6. Copy the custom identity keystore file, say oimsupporttrust.jks, under the *DOMAIN_HOME*/config/fmwconfig/ directory. Enter the absolute path of this key store (*DOMAIN_HOME*/config/fmwconfig/oimsupporttrust.jks) in the Custom Identity Keystore field.

Note:

- The trust keystore created at *DOMAIN_HOME*/config/fmwconfig/ by Oracle Identity Manager during installation is default-keystore.jks.
- If you are using a different name for truststore than the default name, which is default-keystore.jks, then perform the following steps:

    1. Add Oracle Identity Manager Credential store map key. If you are using any other name, such as oimsupporttrust.jks, then create a key in the credential store by using Oracle Enterprise Manager as default-keystore.jks is created with Oracle Identity Manager configuration by default. To create a key in the credential store:
        a. Login to Oracle Enterprise Manager.
        b. Expand **Weblogic Domain**, ***DOMAIN_NAME***. Right-click ***DOMAIN_NAME***, and select **Security, Credentials**.
        c. In the Credential Store Provider table, click **oim**.
        d. Create a key with type as **Password** and with the credentials, User Name: `oimsupporttrust.jks`, Password: *`password`*.
    2. Change DirectDB, SSLConfig config in the oim-Config.xml file either by exporting/importing this file from MDS or by using Enterprise Manager. For the latter, navigate to **oracle.iam**, **XMLConfig**, **DirectDB**, **SSLConfig** in Application Defined MBeans section of System Mbean Browser, and then change the SSL parameters, for example:

```
SSLConfig
dBTrustStore="oimsupporttrust.jks"g
SSLConfig DBTrustStorePasswordKey =
NAME_OF_CSF_KEY
```

7. Specify JKS as the custom identity keystore type.
8. Type the password (*`password`*) into the Custom Identity Keystore Passphrase and the Confirm Custom Identity Keystore Passphrase fields.

Note: If you are creating a custom trust keystore, then perform the steps 6 to 8 of this section for custom trust keystore field as well.

9. Click **Save**.
10. Click the **SSL** tab.
11. Type `supportpvtkey` as the private key alias.
12. Type the password (*password*) into the Private Key Passphrase and the Confirm Private Key Passphrase fields.
13. Click **Save**.
14. Perform similar steps (steps 1 through 13) for Admin and SOA Servers.
15. Click **Activate changes**.
16. Import the certificate that you exported in "Exporting the Certificate" into the SPML client truststore and Java Standard Trust Store, and WebLogic trust store:

    ```
    MW_HOME/wlserver_10.3/server/lib/cacerts
    ```

    For example:

    ```
    ./keytool -importcert -alias startssl -keystore
    MW_HOME/wlserver_10.3/server/lib/cacerts -storepass <password> -file
    supportpvtkeycert.pem
    ```

    ```
    JAVA_HOME/jre/lib/security/cacerts
    ```

    For example:

    ```
    ./keytool -importcert -alias startssl -keystore
    JAVA_HOME/jre/lib/security/cacerts -storepass <password> -file
    supportpvtkeycert.pem
    ```

    Note: Where `<password>` is the default password for Java's Standard truststore (*JAVA_HOME*/jre/lib/security/cacerts).

    See "Importing the Certificate" for information about importing the certificate.

    Note: If the CN of the certificate is not the same as the hostname of the machine where WLS is installed, then you need to select the hostname verification as None. To do so, go to SSL tab, Advanced section, select **None** from the Hostname Verification list.

25.3.3 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
- Changing OimFrontEndURL to Use OIM SSL Port
- Changing backOfficeURL to Use SOA SSL Port
- Changing SOA Server URL to Use SOA SSL Port

25.3.3.1 Enabling SSL for Oracle Identity Manager

Enabling SSL for Oracle Identity Manager is described in the following sections:

- Enabling SSL for Oracle Identity Manager By Using Default Setting

- Enabling SSL for Oracle Identity Manager By Using Custom Keystore

25.3.3.1.1 Enabling SSL for Oracle Identity Manager By Using Default Setting

To enable SSL for Oracle Identity Manager and SOA servers by using default setting:

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.

Note:

- If JDK 7u40 or later is used, then the value of the keysize option must be greater than or equal to 1024. For more information about this limitation, see "Default x.509 Certificates Have Longer Key Length" at the following URL:

  http://www.oracle.com/technetwork/java/javase/7u40-relnotes-2004172.html

- If JDK 7u40 or later is used and SSL is configured by using the default certificates as described in "Enabling SSL for Oracle Identity Manager By Using Default Setting" on page 25-27, then apply patch 13964737. You can download this patch from the My Oracle Support web site at:

  https://support.oracle.com

25.3.3.1.2 Enabling SSL for Oracle Identity Manager By Using Custom Keystore

To enable SSL for Oracle Identity Manager by using custom keystore:

1. In the *DOMAIN_HOME*/bin/setDomainEnv.sh file for UNIX or *DOMAIN_HOME*\bin\setDomainEnv.cmd for Microsoft Windows. Locate the line # SET THE CLASSPATH and add the following:

```
TLS_JAVA_OPTIONS=" -Dweblogic.management.username=username
-Dweblogic.management.password=password
-Djavax.net.ssl.trustStore=$TRUSTSTORE_LOCATION
-Dweblogic.security.SSL.ignoreHostnameVerification=true
-Dweblogic.security.SSL.enforceConstraints=off -Dweblogic.ssl.JSSEEnabled=true
-Dweblogic.security.SSL.enableJSSE=true
-Dweblogic.security.SSL.protocolVersion=TLSv1.2 -Dhttps.protocols=TLSv1.2
-Djdk.tls.client.protocols=TLSv1.2
-Djdk.tls.disabledAlgorithms=SSLv2Hello,SSLv3,TLSv1,TLSv1.1 -Dssl.debug=true
-Djavax.net.debug=ssl:handshake:verbose "

JAVA_OPTIONS="${JAVA_OPTIONS} ${TLS_JAVA_OPTIONS}"
export JAVA_OPTIONS
```

   Here, the value of TRUSTSTORE_LOCATION in case of custom trust store is:

```
DOMAIN_HOME/config/fmwconfig/oimsupporttrust.jks
```

   The value of TRUSTSTORE_LOCATION in case of Demo trust store is:

```
${WL_HOME}/server/lib/DemoTrust.jks
```

The value of TRUSTSTORE_LOCATION in case of Java Standard Trust store is:

```
$JAVA_HOME/jre/lib/security/cacerts
```

Note:
- These settings work with JDK7u131, Use `-Dssl.debug=true -Djavax.net.debug=ssl:handshake:verbose` only for enabling SSL debugging information.
- Stop Weblogic.sh is not supporting to pass or set the trust store in use. These scripts use Java standard trust. Import certificate in Java standard trust along with custom trust store while doing the basic SSL configurations.

---

2. In a text editor, open the startManagedWebLogic.sh file and do the following:
   1. Change the value of ADMIN_URL to point to a SSL URL. For example:

      ADMIN_URL="https://myhost.mycompany.com:7002"
   2. Comment out below line:

      ```
      #JAVA_OPTIONS="-Dweblogic.security.SSL.trustedCAKeyStore="WL_HOME/server/l
      ib/cacerts" ${JAVA_OPTIONS}"
      #export JAVA_OPTIONS
      ```

   Save the startManagedWebLogic.sh file.
3. Restart all servers for the changes to take effect.

   Ensure that when only SSL listen port is enabled on Oracle Identity Manager server and non-SSL listen port is disabled, you must set the value of the providerURL JVM system property to point to the Oracle Identity Manager RMI t3s URL, as follows:

   ```
   -DproviderURL=t3s://OIM_HOST:OIM_SSL_PORT
   ```

   This can be done by setting the value of the JAVA_OPTIONS environment variable before starting Oracle Identity Manager Managed Server from the command prompt.

   For instance, on Linux, if you are using csh shell, then set the environment variable in the following way:

   ```
   setenv JAVA_OPTIONS -DproviderURL=t3s://OIM_HOST:OIM_SSL_PORT
   ```

   For bash, set the following:

   ```
   export JAVA_OPTIONS=-DproviderURL=t3s://OIM_HOST:OIM_SSL_PORT
   ```

   On Microsoft Windows, set the environment variable in the following way:

   ```
   SET JAVA_OPTIONS=-DproviderURL=t3s://OIM_HOST:OIM_SSL_PORT
   ```

   If Oracle Identity Manager server is managed through Node Manager, then add the following as an argument under oim_server, Configuration, Server start, Arguments.

   ```
   -DproviderURL=t3s://OIM_HOST:OIM_SSL_PORT
   ```

Optionally, you can set the parameters in step 17 to startWeblogic.sh as well as startmanagedWeblogic.sh to start the server via these scripts.

Backup the WL_HOME/common/nodemanager/nodemanager.properties file. Open the file and add the following:

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreType=JKS
CustomIdentityKeyStoreFileName=DOMAIN_HOME/config/fmwconfig/oimsupporttrust.jk
s
CustomIdentityAlias=supportpvtkey
CustomIdentityKeyStorePassPhrase=password
CustomIdentityPrivateKeyPassPhrase=privatepassword
```

Ensure that the path, alias, and password is updated as per the AdminServer configuration.

Note: Oracle Identity Manager can connect to SOA via web services. If web service invocation fails, then SOA cannot connect to Oracle Identity Manager, and as a result, requests can be stuck. For example, after a create user request is approved, the request might be stuck because the corresponding SOA composite is not able to invoke the request web service deployed on Oracle Identity Manager server, which is SSL-enabled. To avoid such issues, set JAVA_OPTIONS in the in setDomainEnv.sh file, for example, with:

```
-Djavax.net.ssl.trustStore==DOMAIN_HOME/config/fmwconfig/oimsupport
trust.jks
```

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

- Changing OimFrontEndURL to Use OIM SSL Port
- Changing backOfficeURL to Use SOA SSL Port
- Changing SOA Server URL to Use SOA SSL Port

25.3.3.2 Changing OimFrontEndURL to Use OIM SSL Port

To change the OimFrontEndURL to use OIM 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, and then oim (11.1.2.0.0).
3. Right click and select System MBean Browser.
4. Under Application Defined MBeans, navigate to **oracle.iam**, **Server:<oim_servername>, Application:oim**, **XMLConfig**, **Config**, **XMLConfig.DiscoveryConfig**, **Discovery**.

In a clustered deployment, when you select **oracle.iam** under Application Defined MBeans, Oracle Identity Manager server name is displayed. Select the server and continue with the navigation.

Note: In a clustered deployment, the change to the OimFrontEndURL must be made on each server in the cluster.

5. Enter a new value for the "OimFrontEndURL" attribute and click **Apply** to save the changes.

   For example:

   ```
   https://myoimserver.mycompany.com:14002
   ```

Note: Fusion Apps or SPML clients store Oracle Identity Manager URL for invoking SPML and also send callback response. Therefore, there are changes needed corresponding to this. Also, if Oracle Identity Manager is integrated with OAM/OAAM/OIN, there may be corresponding changes necessary.

25.3.3.3 Changing backOfficeURL to Use SOA SSL Port

To change the backOfficeURL to use SOA 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:

   ```
   t3s://mywls1.mycompany.com:8002
   ```

   ```
   t3s://mywls1.mycompany.com:8002,mywls2.mycompany.com:8003
   ```

25.3.3.4 Changing SOA Server URL to Use SOA SSL Port

To change SOA server URL to use SOA SSL port:

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

   For example:

   ```
   http://ADMINISTRATIVE_SERVER/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 of the Rmiurl attribute.

   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:

   ```
   t3s://mysoa1.mycompany.com:8002
   ```

   ```
   t3s://mysoa1.mycompany.com:8002,mysoa2.mycompany.com:8003,mysoa3.mycompany.com:8004
   ```

6. Change the value of the Soapurl attribute. For example:

   ```
   https://mysoa.mycompany.com:8002
   ```

   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.

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

The SOA server URL must be enabled in ForeignJNDIProvider-SOA as well:

1. Login to WebLogic Administrative Console.
2. Navigate to **domain**, **services**, **ForeignJNDIProvider**.
3. Click **ForeignJNDIProvider-SOA**, and modify it to:

   ```
   t3s://HOST_NAME:SSL_SOA_PORT
   ```

   For example:

   ```
   t3s://mysoa.mycompany.com:8002
   ```

25.3.4 Enabling SSL for Oracle Identity Manager DB

You need to perform the following configurations to enable SSL for Oracle Identity Manager DB:

- Creating KeyStores and Certificates
- Setting Up DB in Server-Authentication SSL Mode
- Updating Oracle Identity Manager
- Updating WebLogic Server

25.3.4.1 Creating KeyStores and Certificates

You can create server side and client side KeyStores using the orapki utility. This utility is 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 KEYSTORE_PASSWORD
   ```

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 KEYSTORE_PASSWORD -sign_alg sha256
   ```

4. View the wallet using the command:

   ```
   ./orapki wallet display -wallet CA_keystore.p12 -pwd KEYSTORE_PASSWORD
   ```

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 KEYSTORE_PASSWORD
   ```

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
   KEYSTORE_PASSWORD
   ```

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
   KEYSTORE_PASSWORD -sign_alg sha256
   ```

3. Export the certificate request to a file, which is 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 KEYSTORE_PASSWORD -sign_alg sha256
```

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 KEYSTORE_PASSWORD -sign_alg sha256
```

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 KEYSTORE_PASSWORD -sign_alg sha256
```

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 KEYSTORE_PASSWORD -sign_alg sha256
```

Creating Client Side Wallet

To create a client side (Oracle Identity Manager server) wallet:

1. Create a client keystore or use existing keystore default-keystore.jks at 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:

```
JAVA_HOME/jre/bin/keytool -import -trustcacerts -alias dbtrusted -noprompt
-keystore default-keystore.jks -file self_signed_CA.cert -storepass
KEYSTORE_PASSWORD
```

   Here, *KEYSTORE_PASSWORD* is the password given for the keystore during Oracle Identity Manager configuration.

   Note: For custom trust keystore, import the self-signed CA trusted certificate to that, for example:

```
JAVA_HOME/jre/bin/keytool -import -trustcacerts -alias dbtrusted
-noprompt -keystore oimsupporttrust.jks -file self_signed_CA.cert
-storepass KEYSTORE_PASSWORD
```

25.3.4.2 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. Navigate to the path:

   $DB_ORACLE_HOME/network/admin directory

   For example:

   ```
   /u01/app/user1/product/12.1.0/dbhome_1/network/admin
   ```

3. Configuring the listener.ora file as follows:
   a. 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: DB_HOME/listener.ora
      # Generated by Oracle configuration tools.

      SSL_VERSION = 1.2
      SSL_CLIENT_AUTHENTICATION = FALSE

      WALLET_LOCATION =
        (SOURCE =
          (METHOD = FILE)
          (METHOD_DATA =
            (DIRECTORY = DB_HOME/server_keystore_ssl.p12)
          )
        )

      LISTENER =
        (DESCRIPTION_LIST =
          (DESCRIPTION =
            (ADDRESS = (PROTOCOL = TCPS)(HOST = server1.mycompany.com)(PORT =
      2484))
          )
          (DESCRIPTION =
            (ADDRESS = (PROTOCOL = TCP)(HOST = server1.mycompany.com)(PORT =
      1521))
          )
        )

      TRACE_LEVEL_LISTENER = SUPPORT
      ```

4. Configure the sqlnet.ora file as follows:
   a. 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: DB_HOME/sqlnet.ora
      ```

```
# Generated by Oracle configuration tools.

SQLNET.AUTHENTICATION_SERVICES= (BEQ, TCPS)

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY =
/u01/app/user1/product/12.1.0/dbhome_1/bin/server_keystore_ssl.p12)
    )
  )

SQLNET.AUTHENTICATION_SERVICES = (TCPS,NTS,BEQ)
SSL_CLIENT_AUTHENTICATION = FALSE
```

5. Configure the tnsnames.ora file as follows:

   a. 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: DB_HOME/tnsnames.ora
# Generated by Oracle configuration tools.
ORCL12C =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCPS)(HOST = server1.mycompany.com))(PORT =
2484))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = proddb)
    )
  )

LISTENER_ORCL12C =
  (ADDRESS = (PROTOCOL = TCPS)(HOST = server1.mycompany.com)(PORT = 2484))
```

6. Start/Stop utilities for DB server.
7. Start the DB server.

25.3.4.3 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. Log in to Enterprise Manager.
2. Navigate to Identity and Access, OIM.
3. Right click and navigate to System MBean Browser.
4. Under Application Defined MBeans, navigate to oracle.iam, Application:oim, XMLConfig, Config, XMLConfig.DirectDBConfig, and DirectDB.
5. 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:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=T
CPS)(HOST=server1.mycompany.com)(PORT=2484)))(CONNECT_DATA=(SERVI
CE_NAME=server12c.mycompany.com))(SECURITY=(SSL_SERVER_CERT_DN="
CN=root_test,C=US")))
```

6. Restart the Oracle Identity Manager server.

25.3.4.4 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:

- Updating Datasource oimOperationsDB Configuration
- Updating Oracle Identity Manager Authenticators

> **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 oimOperationsDB Configuration

To update the Change Datasource oimOperationsDB Configuration:

1. Log in to WebLogic Server.
2. Navigate to **Services**, **JDBC**, **Data Sources**, **oimOperationsDB**.
3. Click the **Connection Pool** tab.
4. Change the value of the URL to reflect the changes to SSl DB host/port, similar to the following example:

   ```
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCPS)(HOST=ser
   ver1.mycompany.com)(PORT=2484)))(CONNECT_DATA=(SERVICE_NAME=server12c.mycompan
   y.com))(SECURITY=(SSL_SERVER_CERT_DN="CN=root_test,C=US")))
   ```

   Where SSL_SERVER_CERT_DN="CN=root_test,C=US" is DB root certificate DN.

5. Update Properties to add the following SSL-related properties:

   ```
   javax.net.ssl.trustStore=DOMAIN_HOME/config/fmwconfig/default-keystore.jks
   javax.net.ssl.trustStoreType=JKS
   javax.net.ssl.trustStorePassword=password
   ```

   Here, *password* is the password given for the keystore during Oracle Identity Manager configuration.

Note: ■Use default-keystore.jks or oimsupporttrust.jks based on values provided for Wallet.

- For custom trust keystore, provide the path of keystore in the javax.net.ssl.trustStore property file. For example:

```
javax.net.ssl.trustStore=DOMAIN_HOME/config/fmwconfig/oimsuppor
ttrust.jks
```

- If required, perform similar updates to all datasources related to SOA, OWSM, or OPSS like ApplicationDB, bip_datasource, EDNDataSource, EDNLocalTxDataSource, mds-oim, mds-owsm, mds-soa, oimJMSStoreDS, opss-DBDS, OraSDPMDataSource, SOADataSource, SOALocalTxDataSource, and soaOIMLookupDB.

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.

25.3.5 Enabling SSL for SOA Approval Composites

To enable SSL for SOA approval composites:

1. Ensure that the SOA Managed Server is running.
2. Log in to Oracle Enterprise Manager by using your WebLogic Server administrator credentials.
3. Expand **SOA**, **soa-infra(soa_server1)**, **default**, and select **DefaultRequestApproval [5.0]**. Then, click **ApprovalTask**, and click the **Administration** tab.
4. Enter a value for the HTTPS port as appropriate, and then click **Apply**.

5. Repeat steps 3 and 4 for each approval composite with a Human Workflow component type, which has a valid worklist URL entry that needs to now use the HTTPS port, for example DefaultOperationalApproval [5.0].

25.3.6 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 Oracle Internet Directory or Oracle Virtual Directory with SSL
- Configuring Oracle Internet Directory
- Configuring Oracle Unified Directory
- Updating Oracle Identity Manager for libOVD details
- Enabling SSL between libOVD and OID/OUD

25.3.6.1 Enabling Oracle Internet Directory or Oracle Virtual Directory with SSL

To enable Oracle Internet Directory or Oracle Virtual Directory with SSL:

1. Log in to the Oracle Internet Directory or Oracle Virtual Directory EM console.
2. Expand **Identity and Access** and navigate to oid or oud1, Administration, Listeners.
3. Click **Create** and enter all the required fields. Create a listener, for example `OIM SSL ENDPOINT`.

   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 Oracle Virtual Directory server for the changes to take effect.

    Note: You must not use the restart option.

25.3.6.2 Configuring Oracle Internet Directory

Configure Oracle Internet Directory with below properties:

1. Set environmental variables by running the below command:

   ```
   setenv PATH /u01/oimhome/Oracle_IDM2/bin:$ORACLE_HOME/bin:$JAVA_HOME/bin:$PATH
   ```

2. Run the below command:

```
orapki wallet create -wallet oim12coidwallet -auto_login
orapki wallet add -wallet oim12coidwallet -dn 'cn=orcladmin' -keysize 2048
-self_signed -validity 3650 -pwd password -sign_alg sha256
```

To export the self signed certificate, run the below command:

```
orapki wallet export -wallet oim12coidwallet -dn 'cn=orcladmin' -cert
oid_self_signed_CA.cert -pwd password
```

3. Verify the properties with below command:

```
ldapsearch -p 3060 -D cn=orcladmin -w password -b
'cn=oid1,cn=osdldapd,cn=subconfigsubentry' -s base objectclass='*' | grep -i
crypto

ldapsearch -p 3060 -D cn=orcladmin -w password -b
'cn=oid1,cn=osdldapd,cn=subconfigsubentry' -s base objectclass='*' | grep -i
ssl
```

4. Verify binding on SSL port is successful with the below command:

```
ldapbind -h server1.mycompany.com -p 3131 -D cn=orcladmin -w password -q -U 2
-W "file:/u01/oidwallet/oim12coidwallet" -P password
```

Note: For more information about Configuring SSL in Oracle Internet Directory, see Configuring Secure Sockets Layer (SSL) in *Administrator's Guide for Oracle Internet Directory*.

25.3.6.3 Configuring Oracle Unified Directory

Configure Oracle Unified Directory with below properties:

While creating Oracle Unified Directory instance, check if SSL is enabled to generate self signed certificate. By default, SSL3, TLS1.1, and TLS1.2 protocols should be enabled.

To disable other protocols and keep only TLS1.2, run the below command:

```
./dsconfig -h localhost -p 1444 -D "cn=oudadmin" -j pwd.txt
set-connection-handler-prop --handler-name "LDAPS Connection Handler" --set
ssl-protocol:TLSv1.2
```

Use below command to check which protocol is used:

```
./dsconfig -h localhost -p 1444 -D "cn=oudadmin" -j pwd.txt
get-connection-handler-prop --handler-name "LDAPS Connection Handler"
```

By default, Admin port number is 1444.

Note: For more information about Configuring SSL in Oracle Unified Directory, see Configuring Security Between Clients and Servers in *Administering Oracle Unified Directory*.

25.3.6.4 Updating Oracle Identity Manager for libOVD details

When LDAPSync is enabled, 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 System Administration.
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 the details.

> **Note:** The Server URL and SSL Server URL must be empty as they are constructed from adapter_os.xml.

5. Ensure that Use SSL is set to true and click **Update**.

25.3.6.5 Enabling SSL between libOVD and OID/OUD

To enable SSL between libOVD and OID/OUD perform the following:

1. To import OID/OUD's certificate to oim trust store and to libOVD keystore:
   a. To Import the exported certificate in OIM trust store, run the command:

```
keytool -import -trustcacerts -alias oidtrusted -noprompt -keystore
oimsupporttrust.jks -file oid_self_signed_CA.cert -storepass password
```

   b. To import the certificate in libOVD keystore, run the command:

```
keytool -import -trustcacerts -alias oidtrusted -noprompt -keystore
adapters.jks -file oid_self_signed_CA.cert -storepass password
```

2. Modify the $DOMAIN_HOME/config/fmwconfig/ovd/CONTEXT/adapters.os_xml file as shown below:
   - Modify the `host port` to `ssl port`.
   - Set `secure` to `true`.
   - Set `protocols` to `TLSv1.2`.
   - Add the below `ciphers` to `cipherSuites`:

```
<cipher>TLS_RSA_WITH_AES_128_CBC_SHA256</cipher>
 <cipher>TLS_RSA_WITH_AES_256_CBC_SHA256</cipher>
 <cipher>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384</cipher>
```

> **Note:** Here CONTEXT is oim. We can use WLST commands to set the parameters as shown below:
>
> ```
> modifyLDAPAdapter(adapterName='CHANGELOG_dir1',
> attribute='Protocols', value='TLSv1.2', contextName='oim')
> ```

25.3.7 Configuring SSL for Design Console

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

1. Copy wlthint3client.jar file from *WEBLOGIC_HOME*/server/lib folder to *DESIGN_CONSOLE_HOME*/ext folder.
2. Edit to replace ./ext/wlfullclient.jar with ./ext/wlthint3client.jar in the relevant file:

For Linux: DESIGN_CONSOLE_HOME/classpath.sh

For Windows: DESIGN_CONSOLE_HOME /classpath.bat

3. Copy *MW_HOME*/modules/cryptoj.jar to the *OIM_HOME*/designconsole/ext/ directory.

4. Edit the $*DESIGN_CONSOLE_HOME*/config/xlconfig.xml file. Make the following changes:

   Change:

```
<Discovery>
            <CoreServer>
<java.naming.provider.url>t3://HOST_NAME:OIM_PORT/oim</java.naming.provider.ur
l>
<java.naming.factory.initial>weblogic.jndi.WLInitialContextFactory</java.namin
g.factory.initial>
            </CoreServer>
</Discovery>
```

   To:

```
<Discovery>
            <CoreServer>
<java.naming.provider.url>t3s://HOST_NAME:OIM_SSL_PORT/oim</java.naming.provid
er.url>
<java.naming.factory.initial>weblogic.jndi.WLInitialContextFactory</java.namin
g.factory.initial>
            </CoreServer>
</Discovery>
```

   Change:

```
<ApplicationURL>http://HOST_NAME:PORT_NUMBER/xlWebApp/loginWorkflowRenderer.do
</ApplicationURL>
```

   To:

```
<ApplicationURL>https://HOST_NAME:OIM_SSL_PORT/xlWebApp/loginWorkflowRenderer.
do</ApplicationURL>
```

5. If $*DESIGN_CONSOLE_HOME*/config/xl.policy does not contain the default grant policy for all, then add the following permission for cryptoj.jar at the end of the file, as shown:

```
grant codeBase "file:DESIGN_CONSOLE_HOME/ext/cryptoj.jar"{
  permission java.security.AllPermission;
};
```

   Copy $*MW_HOME*/modules/cryptoj.jar to the $*OIM_HOME*/designconsole/ext/ directory.

   Note: Here, copying $*MW_HOME*/modules/cryptoj.jar to the $*OIM_HOME*/designconsole/ext/ directory is a mandatory step. Setting the permission is necessary if xl.policy does not contain the default grant policy for all.

6. In the relevant file, add the following properties:

   For Linux: DESIGN_CONSOLE_HOME/xlclient.sh

For Windows: DESIGN_CONSOLE_HOME/xlclient.cmd

```
/u01/jdks/jdk1.7.0_131/bin/java -DXL.ExtendedErrorOptions=TRUE \
   -DXL.HomeDir=. -Djava.security.policy=config/xl.policy \
   -Djava.security.manager
-Djava.security.auth.login.config=config/authwl.conf \
   -Dlog4j.configuration=config/log.properties \
   -DAPPSERVER_TYPE=wls \
   -Djavax.net.ssl.trustStore=$TRUSTSTORE_LOCATION \
   -Dweblogic.security.SSL.protocolVersion=TLSv1.2 \
   -Dhttps.protocols=TLSv1.2 \
   -Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2 \
   -DproviderURL=t3s://server1.mycompany.com:14002 \
   -Dweblogic.ssl.JSSEEnabled=true \
   -Dweblogic.security.SSL.enableJSSE=true \
   -Dweblogic.security.allowCryptoJDefaultJCEVerification=true \
   -Dweblogic.security.SSL.enforceConstraints=off \
   -Dweblogic.security.SSL.ignoreHostnameVerification=true \
   -Dweblogic.StdoutDebugEnabled=true \
   -Dssl.debug=true \
   -Djavax.net.debug=ssl:handshake:verbose \
   -cp $CLASSPATH com.thortech.xl.client.base.tcAppWindow -server server
```

7. Set environment variable TRUSTSTORE_LOCATION to the location of custom/demo/Java Standard trust keystore used at server side.

   For example:

   ```
   setenv TRUSTSTORE_LOCATION DOMAIN_HOME/config/fmwconfig/oimsupporttrust.jks
   ```

Note:

- To get trust store location, in the WebLogic Server Administration Console, click **Environment**, **Servers**. Click OIM_SERVER_NAME to view details of the Oracle Identity Manager server.

  Click KeyStores tab and note down the Trust keystore location in the Trust section.

- If the Design Console and Oracle Identity Manager are deployed on a different host, then copy the Trust keystore to the host on which Design Console is deployed, and set the TRUSTSTORE_LOCATION environment variable to the location where Trust keystore is copied on the local host.

  For example:

  ```
  setenv TRUSTSTORE_LOCATION OIM_HOME/designconsole/DemoTrust.jks
  ```

25.3.8 Configuring SSL for Oracle Identity Manager Utilities with TLS

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

When Oracle Identity Manager is configured with TLS, perform the following steps to configure Oracle Identity Manager utilities:

1. Export the Oracle Identity Manager server certificate and import it into custom keystore oimsupporttrust.jks.

2. Edit the *OIM_HOME*/server/bin/oimClientWrapper.sh file to add the following parameters after $JAVA_HOME/bin/java -cp $CLASSPATH:

```
-Dweblogic.security.SSL.trustedCAKeyStore=$TRUSTSTORE_LOCATION
-Dweblogic.security.SSL.protocolVersion=TLSv1.2 -Dhttps.protocols=TLSv1.2
-Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2
-DproviderURL=t3s://server1.mycompany.com:14002
-Dweblogic.ssl.JSSEEnabled=true -Dweblogic.security.SSL.enableJSSE=true
-Dweblogic.security.allowCryptoJDefaultJCEVerification=true
-Dweblogic.security.SSL.enforceConstraints=off
-Dweblogic.security.SSL.ignoreHostnameVerification=true
-Dweblogic.StdoutDebugEnabled=true -Dssl.debug=true
-Djavax.net.debug=ssl:handshake:verbose
```

3. Before running the utilities, in the command prompt, set the TRUSTSTORE_LOCATION environment variable to pointing towards the location of custom/demo/Java Standard trust keystore used at server side. For example:

```
setenv TRUSTSTORE_LOCATION DOMAIN_HOME/config/fmwconfig/oimsupporttrust.jks
```

> **Note:** Ensure that Oracle Identity Manager server certificate is already imported into above trust store.

4. For clients, such as Remote Manager, and other utilities to connect to Oracle Identity Manager in SSL/TLS way, the public key (certificate) must be made available in the keystore for clients to use it. To do so, export and import public key (certificate) as below:

   a. Export the public certificate from DemoIdentity.jks or oimsupportidentity.jks, which has private keys, by using the following command. Alternatively, you can export from the browser.

   ```
   $JAVA_HOME/jre/bin/keytool -export -file key.cer -alias demoidentity
   -keystore DemoIdentity.jks -storepass DemoIdentityKeyStorePassPhrase
   ```

   In case of custom identity store:

   ```
   $JAVA_HOME/jre/bin/keytool -export -alias supportpvtkey -file
   supportpvtkeycert.pem -keypass password -keystore oimsupportidentity.jks
   -storepass password
   ```

   b. Import that certificate to the client keystore, as shown:

   ```
   $JAVA_HOME/jre/bin/keytool -import -trustcacerts -file key.cer -alias
   qa_certgenca -keystore DemoTrust.jks -storepass
   DemoTrustKeyStorePassPhrase
   ```

   Here, it is DemoTrust.jks for demo keystore or oimsupporttrust.jks for custom key store.

   c. In clients, such as Design Console, Remote Manager, and utilities, point TRUSTSTORE_LOCATION or -Dweblogic.security.SSL.trustedCAKeyStore to this key store, as shown:

   ```
   setenv TRUSTSTORE_LOCATION WL_HOME/server/lib/DemoTrust.jks

   -Dweblogic.security.SSL.trustedCAKeyStore=
   WL_HOME/server/lib/DemoTrust.jks \
   ```

d. To configure SSL using Transport Layer Security (TLS) with additional parameters for the Remote Manager scripts, in a text editor, open the following scripts:

OIM_HOME/remotemanager/remotemanager.sh

Add the following parameters:

```
-Dweblogic.security.SSL.trustedCAKeyStore=$TRUSTSTORE_LOCATION
-Dweblogic.security.SSL.protocolVersion=TLSv1.2 -Dhttps.protocols=TLSv1.2
-Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2
-DproviderURL=t3s://server1.mycompany.com:14002
-Dweblogic.ssl.JSSEEnabled=true -Dweblogic.security.SSL.enableJSSE=true
-Dweblogic.security.allowCryptoJDefaultJCEVerification=true
-Dweblogic.security.SSL. enforceConstraints=off
-Dweblogic.security.SSL.ignoreHostnameVerification=true
-Dweblogic.StdoutDebugEnabled=true -Dssl.debug=true
-Djavax.net.debug=ssl:handshake:verbose
```

Save the changes to the scripts.

26

Securing a Deployment

This chapter contains the following sections:

26.1 Authorizing and Hardening

Securing an Oracle Identity Manager deployment is achieved through authorization and hardening. Authorization controls the access to various components. Hardening secures the components from potential security threats.

Table 26–1 lists the various topics that you can refer for information about securing an Oracle Identity Manager deployment:

Table 26–1 Securing a Deployment

Topic	Topic Type	Information Covered
"Managing the Scheduler" on page 18-1	Hardening	Scheduled tasks and scheduled jobs. Ensure that only required scheduled tasks are enabled.
"System Properties in Oracle Identity Manager" on page 20-1	Hardening	System properties related to system behavior. Ensure that password policies and challenge questions and answers are defined.
"Creating the User Account for Installing Connectors" on page 11-7	Hardening	Specific permissions required to install connectors.
"Configuring Secure Cookies" on page 26-2	Hardening	Enabling Oracle Identity Manager to work over SSL.
"Configuring LDAP Authentication When LDAP Synchronization is Enabled" in the *Oracle Fusion Middleware Integration Guide for Oracle Identity Management Suite*	Hardening	Enabling LDAP authentication.
"URL Changes Related to Oracle Identity Manager" on page 25-1	Hardening	Steps to make the corresponding changes in Oracle Identity Manager and Oracle WebLogic configuration for any change in the integrated and dependent applications
"Password Changes Related to Oracle Identity Manager" on page 25-9	Hardening	Steps to make the changes to the password in Oracle Identity Manger and Oracle WebLogic configuration for any change in the dependent or integrated products.
"Configuring SSL for Oracle Identity Manager" on page 25-20	Hardening	Securing Oracle Identity Manager by configuring SSL.

Table 26–1 (Cont.) Securing a Deployment

Topic	Topic Type	Information Covered
"Managing Password Policies" in the *Performing Self Service Tasks with Oracle Identity Manager*.	Hardening	Password policy configuration.
"Security Architecture" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*	Authorization	Authorization and security model in Oracle Identity Manager
"Check Permissions for Roles" in the *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*	Authorization	Permissions for role while importing and exporting roles. Check for any errors in setting data object permissions if data object is missing.

See Also: *Oracle Fusion Middleware Installation Guide for Oracle Identity and Access Management* and *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management* for information about Oracle Identity Management software integrations and related security aspects

26.2 Configuring Secure Cookies

Oracle Identity Manager application is not configured for SSL access by default. So, the `oimjsessionid` cookie used by Oracle Identity Manager web applications is not secure for HTTPS access. In other words, the `cookie-secure` tag is not set to `true`. However, when SSL access to Oracle Identity Manager is enabled, it is recommended to configure `oimjsessionid` as a secure cookie by setting the `cookie-secure` tag to `true`. This tag enables the browser to send the cookie back over an HTTPS connection only. This ensures that the cookie ID is secure and is only used upon HTTPS access of Oracle Identity Manager. This also implies that HTTP access to Oracle Identity Manager no longer works when this feature is enabled. In addition, the `url-rewriting-enabled` element must be disabled.

Secure cookies need to be configured for the following Oracle Identity Manager UI pages:

- /identity, available in *OIM_HOME*/apps/oracle.iam.console.identity.self-service.ear/oracle.iam.console.identity.self-service.war
- /sysadmin, available in *OIM_HOME*/apps/oracle.iam.console.identity.sysadmin.ear/oracle.iam.console.identity.sysadmin.war
- /oim, available in *OIM_HOME*/apps/oim.ear/iam-consoles-faces.war
- /xlWebApp, available in *OIM_HOME*/apps/oim.ear/xlWebApp.war

Secure cookies can be configured by updating the deployment plan for each of the applications, which are `iam.console.identity.self-service.ear`, `oracle.iam.console.identity.sysadmin.ear`, and `oim.ear`.

This section describes how to configure secure cookies in the default scenario when there is no deployment plan for these applications. It also describes the configuration when updating a current deployment plan if you have explicitly configured it. This section contains the following topics:

- Configuring a New Deployment Plan

- Updating an Existing Deployment Plan

26.2.1 Configuring a New Deployment Plan

Deployment plan specific to the applications can be configured by logging into the WebLogic Administrative Console. The following are sample deployment plans with secure cookie enabled for each of the applications:

- Following is the sample deployment plan XML for the `oracle.iam.console.identity.self-service.ear` application. In this deployment plan, `cookie-secure` is configured to `true`, and `url-rewriting-enabled` is configured to `false` for the `oracle.iam.console.identity.self-service.war` web application:

```
<?xml version='1.0' encoding='UTF-8'?>
<deployment-plan xmlns="http://xmlns.oracle.com/weblogic/deployment-plan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/deployment-plan
http://xmlns.oracle.com/weblogic/deployment-plan/1.0/deployment-plan.xsd">


<application-name>oracle.iam.console.identity.self-service.ear#V2.0</applicati
on-name>
  <variable-definition>
    <variable>
      <name>SessionDescriptor_CookieSecure_identity_13909448828172</name>
      <value>true</value>
    </variable>
    <variable>

<name>SessionDescriptor_UrlRewritingEnabled_identity_139095392691833</name>
      <value>false</value>
    </variable>
  </variable-definition>
  <module-override>
    <module-name>oracle.iam.console.identity.self-service.war</module-name>
    <module-type>war</module-type>
    <module-descriptor external="false">
      <root-element>weblogic-web-app</root-element>
      <uri>WEB-INF/weblogic.xml</uri>
      <variable-assignment>
        <name>SessionDescriptor_CookieSecure_identity_13909448828172</name>
        <xpath>/weblogic-web-app/session-descriptor/cookie-secure</xpath>
      </variable-assignment>
      <variable-assignment>

<name>SessionDescriptor_UrlRewritingEnabled_identity_139095392691833</name>

<xpath>/weblogic-web-app/session-descriptor/url-rewriting-enabled</xpath>
      </variable-assignment>
    </module-descriptor>
  </module-override>
</deployment-plan>
```

- The following is the sample deployment plan XML for the `oracle.iam.console.identity.sysadmin.ear` application. In this deployment plan, `cookie-secure` is configured to `true`, and `url-rewriting-enabled` is configured to `false` for the `oracle.iam.console.identity.sysadmin.war` web application.

```
<?xml version='1.0' encoding='UTF-8'?>
<deployment-plan xmlns="http://xmlns.oracle.com/weblogic/deployment-plan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/deployment-plan
http://xmlns.oracle.com/weblogic/deployment-plan/1.0/deployment-plan.xsd">


<application-name>oracle.iam.console.identity.sysadmin.ear#V2.0</application-n
ame>
  <variable-definition>
    <variable>
      <name>SessionDescriptor_CookieSecure_sysadmin_13909448828173</name>
      <value>true</value>
    </variable>
    <variable>

<name>SessionDescriptor_UrlRewritingEnabled_sysadmin_139095392691834</name>
      <value>false</value>
    </variable>
  </variable-definition>
  <module-override>
    <module-name>oracle.iam.console.identity.sysadmin.war</module-name>
    <module-type>war</module-type>
    <module-descriptor external="false">
      <root-element>weblogic-web-app</root-element>
      <uri>WEB-INF/weblogic.xml</uri>
      <variable-assignment>
        <name>SessionDescriptor_CookieSecure_sysadmin_13909448828173</name>
        <xpath>/weblogic-web-app/session-descriptor/cookie-secure</xpath>
      </variable-assignment>
      <variable-assignment>

<name>SessionDescriptor_UrlRewritingEnabled_sysadmin_139095392691834</name>

<xpath>/weblogic-web-app/session-descriptor/url-rewriting-enabled</xpath>
      </variable-assignment>
    </module-descriptor>
  </module-override>
</deployment-plan>
```

- The following is the sample deployment plan XML for the `oim.ear` application. In this deployment plan, `cookie-secure` is configured to `true`, and `url-rewriting-enabled` is configured to `false` for the `iam-consoles-faces.war` and `xlWebApp.war` web applications.

```
<?xml version='1.0' encoding='UTF-8'?>
<deployment-plan xmlns="http://xmlns.oracle.com/weblogic/deployment-plan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/deployment-plan
http://xmlns.oracle.com/weblogic/deployment-plan/1.0/deployment-plan.xsd">

  <application-name>oim#11.1.2.0.0</application-name>
  <variable-definition>
    <variable>
      <name>SessionDescriptor_CookieSecure_oim_13909448828170</name>
      <value>true</value>
    </variable>
    <variable>
      <name>SessionDescriptor_UrlRewritingEnabled_oim_139095392691831</name>
      <value>false</value>
```

```
    </variable>
    <variable>
      <name>SessionDescriptor_CookieSecure_xlWebApp_13909448828171</name>
      <value>true</value>
    </variable>
    <variable>

<name>SessionDescriptor_UrlRewritingEnabled_xlWebApp_139095392691832</name>
      <value>false</value>
    </variable>
  </variable-definition>
  <module-override>
    <module-name>iam-consoles-faces.war</module-name>
    <module-type>war</module-type>
    <module-descriptor external="false">
      <root-element>weblogic-web-app</root-element>
      <uri>WEB-INF/weblogic.xml</uri>
      <variable-assignment>
        <name>SessionDescriptor_CookieSecure_oim_13909448828170</name>
        <xpath>/weblogic-web-app/session-descriptor/cookie-secure</xpath>
      </variable-assignment>
      <variable-assignment>
        <name>SessionDescriptor_UrlRewritingEnabled_oim_139095392691831</name>

<xpath>/weblogic-web-app/session-descriptor/url-rewriting-enabled</xpath>
      </variable-assignment>
    </module-descriptor>
  </module-override>
  <module-override>
    <module-name>xlWebApp.war</module-name>
    <module-type>war</module-type>
    <module-descriptor external="false">
      <root-element>weblogic-web-app</root-element>
      <uri>WEB-INF/weblogic.xml</uri>
      <variable-assignment>
        <name>SessionDescriptor_CookieSecure_xlWebApp_13909448828171</name>
        <xpath>/weblogic-web-app/session-descriptor/cookie-secure</xpath>
      </variable-assignment>
      <variable-assignment>

<name>SessionDescriptor_UrlRewritingEnabled_xlWebApp_139095392691832</name>

<xpath>/weblogic-web-app/session-descriptor/url-rewriting-enabled</xpath>
      </variable-assignment>
    </module-descriptor>
  </module-override>
</deployment-plan>
```

To configure the deployment plan(s), copy them to the host on which the Oracle Identity Manager application is deployed. Perform the following steps for all the applications, which as `iam.console.identity.self-service.ear`, `oracle.iam.console.identity.sysadmin.ear`, and `oim.ear`:

1. Login to WebLogic Administrative Console.
2. Navigate to **Deployments**, and then select the application.
3. Click **Update**. The Update Application Assistant page is displayed.
4. Click **Change Path** against the deployment plan path configuration.

5. Specify the path to the deployment plan XML file specific to the application, and click **Next**.
6. Select the **Update this application in place with new deployment plan changes** option. Click **Finish** to complete the deployment plan configuration. Activate changes if required.

Note: You can ignore the following error while updating the deployment plan for `iam.console.identity.self-service.ear` and `oracle.iam.console.identity.sysadmin.ear`:

```
'weblogic.management.DeploymentException:  The application
oracle.iam.console.identity.self-service.ear#V2.0 cannot have the
resource WEB-INF/weblogic.xml updated dynamically. Either:
1.) The resource does not exist.
 or
2) The resource cannot be changed dynamically.
```

---

7. Perform steps 1 through 6 for all the three applications.
8. Restart the Oracle Identity Manager Managed Server.

26.2.2 Updating an Existing Deployment Plan

If any of the applications, `iam.console.identity.self-service.ear`, `oracle.iam.console.identity.sysadmin.ear`, and `oim.ear` have an existing deployment plan, then you must update it to configure `cookie-secure` and `url-rewriting-enabled`. To do so, locate the corresponding deployment plan XML file, and edit it to add the highlighted content (in bold), as shown in the sample deployment plans in "Configuring a New Deployment Plan" on page 26-3.

For example, to configure `cookie-secure` for `oracle.iam.console.identity.self-service.war` web application, add the highlighted content as follows:

```
<?xml version='1.0' encoding='UTF-8'?>
<deployment-plan xmlns="http://xmlns.oracle.com/weblogic/deployment-plan"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.oracle.com/weblogic/deployment-plan
http://xmlns.oracle.com/weblogic/deployment-plan/1.0/deployment-plan.xsd">


<application-name>oracle.iam.console.identity.self-service.ear#V2.0</application-n
ame>
  ..........
  ..........
  <variable-definition>
   ..........
    <variable>
      <name>SessionDescriptor_CookieSecure_identity_13909448828172</name>
      <value>true</value>
    </variable>
    <variable>
      <name>SessionDescriptor_UrlRewritingEnabled_identity_139095392691833</name>
      <value>false</value>
    </variable>
    ..........
  </variable-definition>
```

```
  ……
  ……
  <module-override>
    <module-name>oracle.iam.console.identity.self-service.war</module-name>
    <module-type>war</module-type>
    <module-descriptor external="false">
      <root-element>weblogic-web-app</root-element>
      <uri>WEB-INF/weblogic.xml</uri>
      ……..
      <variable-assignment>
        <name>SessionDescriptor_CookieSecure_identity_13909448828172</name>
        <xpath>/weblogic-web-app/session-descriptor/cookie-secure</xpath>
      </variable-assignment>
      <variable-assignment>

<name>SessionDescriptor_UrlRewritingEnabled_identity_139095392691833</name>
        <xpath>/weblogic-web-app/session-descriptor/url-rewriting-enabled</xpath>
      </variable-assignment>
      ……
    </module-descriptor>
  </module-override>
</deployment-plan>
```

Save the updated the deployment plan XML file, and then restart the Oracle Identity Manager Managed Server for the changes to take effect.

Part X

Diagnostics and Troubleshooting

This part describes diagnostics in Oracle Identity Manager and troubleshooting tasks.

It contains the following chapter:

- Chapter 27, "Using Enterprise Manager for Managing Oracle Identity Manager"

27

Using Enterprise Manager for Managing Oracle Identity Manager

This chapter describes how to configure Oracle Identity Manager using Oracle Enterprise Manager. It contains the following sections:

- Managing Oracle Identity Manager Configuration
- Using the OrchestrationEngine MBean
- Configuring Logging

27.1 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:

- Using MBeans for Configuration Changes
- Exporting and Importing Configuration Files

27.1.1 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 Oracle Enterprise Manager Fusion Middleware Control by using the URL in the following format:

   http://*ADMINSTRATION_SERVER*:*PORT*/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.

27.1.2 Exporting and Importing Configuration Files

To export or import configuration files:

1. When the administrative server and at least one Oracle Identity Manager managed server is running, login to Oracle Enterprise Manager Fusion Middleware Control by using the URL in the following format:

   http://*ADMINSTRATION_SERVER*:*PORT*/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:OIMMetadata**, **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.

27.2 Using the OrchestrationEngine MBean

You can manage the orchestration engine by using the OrchestrationEngine MBean provided by Oracle Enterprise Manager. This section describes the MBean and its various operations and parameters. It contains the following topics:

- Accessing the OrchestrationEngine MBean
- Understanding the Operations Supported by the MBean
- Diagnosing Operation Failures Using the Orchestration Engine

27.2.1 Accessing the OrchestrationEngine MBean

To access the OrchestrationEngine MBean:

1. Login to Oracle Enterprise Manager.
2. Click **WebLogic Domain** at the top, and select **System MBean Browser**.

3. Expand **Application Defined Mbeans**, **oracle.iam**, **Server:*SERVER_INSTANCE_NAME***, **Application:oim**, **Kernel**, and then click **OrchestrationEngine**.

   The Operations tab of the Application Defined MBeans: Kernel:OrchestrationEngine page is displayed.

4. You can expand **Show MBean Information** to display the details of the OrchestrationEngine MBean, specifically the full MBean name and description.

   The Operations tab displays in a tabular format the operation names, descriptions, parameters, and return types of the operations that you can invoke by using the OrchestrationEngine MBean.

5. Click an operation name to open the details of the operation. The operation details page displays the full MBean name, operation name, description of the operation, and return type. It also lists the parameters and allows you to enter values for the parameters.

27.2.2 Understanding the Operations Supported by the MBean

Table 27–1 lists the operations supported by the OrchestrationEngine MBean.

Table 27–1 Operations Supported by OrchestrationEngine

Operation	Description
dump	This operation dumps complete orchestrations to the console or to a file. The console cannot print more then three to five orchestrations because of size limitations, and therefore, file system must be used. Paging must be used to dump data in chunks if the number of processes to be dumped is high, as it can put load on the server. If paging is used to dump data to a single file, then set the appendFile parameter to true, so that page dumps are appended in the file. Otherwise, use separate file per page dump.
familyTree	This operation returns the entire family tree of a process whose ID is provided as a parameter. The family includes children, siblings, and parent up to n-level. Setting the value of the `detailed` parameter to `true` returns full detailed process, which is a heavyweight call for the server, and must be used with care.
findEventHandlers	This operation returns a list of supported event handlers for a particular combination of entity type and operation. The parameters of this operation are case-sensitive. Therefore, when some custom handlers are defined, it helps in debugging if they have the right case as these values are used in case-sensitive manner. All the handlers are returned in the order of execution.
findEventsForProcess	This operation returns the actual list of events applicable in the context/flow for a process, whose ID and/or name is provided. Handlers are returned in the order of execution. The list of handlers is not the complete handler list of an entity type and operation.

Table 27–1 (Cont.) Operations Supported by OrchestrationEngine

Operation	Description
findOperations	This operation returns a list of all the configured operations for an entity type. If nothing is provided as parameter, then it returns a complete list of operations across entity operations.
findProcess	This operation returns a list of processes that satisfies the criteria based on the parameters passed. If search is not ID-based, then `pageSize` and `pageNumber` must be used to process chunks of data as the call is heavy. Setting the `detailed` parameter to `true` returns a full detailed orchestration object. Therefore, this parameter must used with care to prevent extra load on the server. If no value is provided as parameter value, then it returns all the processes saved in the database, which can be a large number.
listEntityTypes	This operation returns a list of Oracle Identity Manager entity types.

27.2.3 Diagnosing Operation Failures Using the Orchestration Engine

Most of the operations done on various entities in Oracle Identity Manager go via the orchestration engine. The list of entity types using orchestration engine as their back bone can be obtained via the `listEntityTypes` operation on the OrchestrationEngine Mbean.

End to end detailed flow of every operation is logged in the log files. For debugging purposes, finer details can be obtained by setting the logging level of the `oracle.iam.platform.kernel` logger to `INFO` or `FINE`.

Orchestrations only get serialized in the database if they have not achieved completed status, which might occur because of failures or waiting for another thread to resume processing.

To understand the cause of incomplete processing of any orchestration process, which can be because of various reasons, you can either look into the logs or use the orchestration process ID obtained from logs to get the details from the OrchestrationEngine Mbean.

> **Note:** Orchestration Process ID is a unique combination of two fields, a long type ID and a string type Name. Either of the two can be provided to the Mbean operations to get results. You can provide both for exact record match.

Orchestration process ID can be obtained in the following ways:

- From the log files
- From `getOrchestrationIds` operation of EventDiagnostic Mbean (`oracle.iam:Location=oim_server1,name=EventDiagnostic,type=Reconciliation,Application=oim`) for reconciliation flows.
- From the `processInfo` operation of the RequestDiagnostic MBean (`oracle.iam:Location=oim_server1,name=RequestDiagnosticMXBean,type=IAMAppRuntimeMBean,Application=oim`), by providing the request ID or from the orchestration_process_id column of the request table for request flows.
- By using the findProcess operation of the OrchestrationEngine MBean, which searches through the database of incomplete orchestrations based on the provided criteria

Note: If the findProcess operation of the MBean for a particular process ID returns nothing, then it means that either the provided ID is incorrect or the particular process completed successfully and does not exist in the database. Information for such a process ID is available only in the log files.

After the process ID is found, perform the following steps to diagnose operation failures:

1. Invoke the `findProcess` operation of the MBean, pass the process ID and set the `detailed` parameter as `true`. This provides all the internal details of the process.
2. Get the details of the handlers involved in the order of execution by invoking the `findEventsForProcess` operation on the MBean.
3. Dump the complete process to the console or a file by invoking the `dump` operation of the MBean. Pass the process ID and the file name if it is required to be dumped to a file.
4. Complex cases involving parent and child orchestrations up to n-level can be completely traced by invoking the `familyTree` operation on the OrchestrationEngine MBean.

If the process ID is not found, then multiple orchestrations can be dumped to a file by using the dump operation of the MBean, based on the parameters provided to the mbean. This dump file along with the log files help understand the cause of various issues. These files can also be provided to Oracle support as part of service request.

27.3 Configuring Logging

Oracle Identity Manager uses two logging services: Oracle Diagnostic Logging (ODL), which is the logging service used by most Oracle Fusion Middleware applications, 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:

- Logging in Oracle Identity Manager By Using ODL
- Logging in Oracle Identity Manager By Using log4j
- Setting Warning State
- Switching Down the Log Level

27.3.1 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.

Logging configuration is controlled by the logging.xml file described in "Log Handler and Logger Configuration" on page 27-7. This file can either be edited directly or edited through the Enterprise Manager. On the Enterprise Manager, the logging configuration can be accessed by clicking the OIM server link and by selecting the

WebLogic Server drop down from the top, and then clicking on Logs - Log Configuration.

To access the logging configuration on the Enterprise Manager:

1. Click the OIM server link.
2. From the WebLogic Server list, select Logs - Log Configuration. All the packages available for logging are displayed on the log configuration screen.

For any additional packages to be logged that are not available in the Enterprise Manager (such as, for connector packages), follow the instructions to manually edit the logging.xml file. The packages specific to Oracle Identity Manager can be accessed under oracle.iam. The different log levels are available for selection under the Oracle Diagnostic Logging Level column. Select a particular log level, and then click **Apply** for the changes to take effect. In addition, new log handlers can be created and configured by clicking the **Log Files** tab.

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 27–3, " 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:

- Message Types and Levels
- Log Handler and Logger Configuration
- Configuring Log Handlers
- Configuring Loggers
- Sample ODL Log Output

27.3.1.1 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 Administering Oracle Identity Manager*. Table 27–2 lists the diagnostic message types that you can use most often with Oracle Identity Manager.

Table 27–2 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.

27.3.1.2 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.

27.3.1.3 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.

27.3.1.3.1 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 Administering Oracle Identity Manager* 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

27.3.1.4 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 27–3 lists the Oracle Identity Manager loggers.

Table 27–3 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.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.
`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.

Table 27–3 (Cont.) Oracle Identity Manager Loggers

Logger	Description
`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.
`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.spmlws` `oracle.iam.wsschema`	Logs events related to web services used for Fusion applications that generate requests for different 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.

Table 27–3 (Cont.) Oracle Identity Manager Loggers

Logger	Description
`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.
`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.

Table 27–3 (Cont.) Oracle Identity Manager Loggers

Logger	Description
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).
oracle.iam.connectors.icfcommon	Provides logging for connector framework.

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.

27.3.1.5 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 Administering Oracle Identity Manager*.

27.3.2 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

27.3.2.1 Log Levels

Table 27–4 lists the log levels for log4j:

Table 27–4 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.

27.3.2.2 Loggers

The loggers for the third-party applications used are:

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

27.3.2.3 Configuring and Enabling Logging

Any of the log levels can be used in the *OIM_HOME*/config/log.properties file for the third-party applications, as follows:

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

27.3.3 Setting Warning State

To set the Oracle Identity Manager server warning state:

1. Set the re-delivery limit on all OIM JMS queues to 1. To do so:
   a. Login to the WebLogic Administration Console as the administrative user.

b. Click **JMS Modules** on the Home page.

c. Click **OIMJMSModule**.

d. Click **Lock & Edit**.

e. For each of the queues, click the queue, and then click the **Delivery Failure** tab. Change the Redelivery Limit value from -1 to 1, and then click **Save**.

f. Make sure you have performed steps 1.d and 1.e for all the queues under OIMJMSModule.

g. Release the configuration and restart Oracle Identity Manager.

This re-delivery is not applicable for existing messages. When the server is restarted, wait for all the good messages to be processed. After that, all the bad messages must be purged.

2. To purge all bad messages:

a. Login to the WebLogic Administration Console as the administrative user.

b. Click **JMS Servers** on the home page.

c. Navigate to **OIMJMSServer**, **Monitoring**, **Active Destinations**.

d. Select the queues that contain messages. Click **Consumption**, **Pause**.

e. Delete the messages, as described in the following URL:

http://docs.oracle.com/cd/E12840_01/wls/docs103/ConsoleHelp/taskhelp/jms_modules/queues/ManageQueues.html

f. After messages are deleted, resume the consumption that has been paused in step 2.d.

3. Restart Oracle Identity Manager.

27.3.4 Switching Down the Log Level

By default, the logging level for `oracle.*` packages is defined as NOTIFICATION/INFO. This results in high volume of log file entries. To avoid this, it is recommended that you switch down the logging level. To do so:

1. In a text editor, open the /domains/*DOMAIN_NAME*/config/fmwconfig/servers/*OIM_SERVER_NAME*/logging.xml file.

2. Search for the following line:

```
<logger name='oracle' level='NOTIFICATION:1'/>
```

3. Change the log level from `NOTIFICATION` to `WARNING`, as shown:

```
<logger name='oracle' level='WARNING:1'/>
```

4. Save the logging.xml file.

Part XI

Appendixes

This part contains the following appendixes:

- Appendix A, "Default User Accounts"
- Appendix B, "Configuring SSO Providers for Oracle Identity Manager"
- Appendix C, "Using Database Roles/Grants for Oracle Identity Manager Database"
- Appendix D, "Enabling Transparent Data Encryption"
- Appendix E, "Troubleshooting Clustered OIM and Eclipselink Cache Coordination"

A

Default User Accounts

Table A lists the default user accounts that are created in Oracle Identity Manager.

Table A–1 Default User Accounts

Account	Description
XELSYSADM	This account is the Oracle Identity Manager administrator (super-user) and is created during installation. You create a password for this account during installation. To change the password at any later point in time after installation, see "Changing Oracle Identity Manager Administrator Password" on page 25-10.
WEBLOGIC	This account is used for integrating SOA and Oracle Identity Manager by using the 'User Role Provider' implementation. When SOA is reconfigured to use LDAP-based user-role provider, Oracle Identity Manager does not require this account. This account is created during installation. You create a password for this account during installation. To change the user name of this account at any later point in time after installation, see "Enabling Oracle Identity Manager to Connect to SOA Using the Administrative Users Provisioned in LDAP" in *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity Management*.
OIMINTERNAL	This account is created during installation and is for internal Oracle Identity Manager use only.

B

Configuring SSO Providers for Oracle Identity Manager

This appendix contains the configuration steps for enabling Oracle Identity Manager for Single Sign On (SSO). To do so, Oracle Identity Manager is enabled to use third-party SSO providers, such as OpenSSO, IBM Tivoli Access Manager, and CA SiteMinder.

This appendix contains the following sections:

B.1 Common Prerequisites for Integration With Third-Party SSO Solutions

This section lists the common prerequisites for integrating Oracle Identity Manager with third-party SSO providers, such as Siteminder, OpenSSO, and Tivoli Access Manager. SSO provider-specific prerequisites are listed separately in corresponding sections. The common prerequisites are as follows:

- Identity population in Oracle Identity Manager is synchronized with identity information in the LDAP registry used by the SSO provider. Oracle Identity Manger's LDAP synchronization feature can be used for this purpose.
- Oracle Identity Manager system administrator (xelsysadm) account should be created in the LDAP repository so that you can perform SSO login to OIM using this administrator account. This account should be created in the same user container that has other OIM users in the LDAP repository. Also ensure that the LDAP user attribute, which is mapped to Oracle Identity Manager user login (uid or samAcountName), has the value set as `XELSYSADM`.
- It is required that the SSO header returned by the SSO provider contains the username value which maps to OIM User Login field.

B.2 Enabling Oracle Identity Manager to Work With OpenSSO

This section contains the following topics:

B.2.1 Prerequisites

The prerequisites for integrating Oracle Identity Manager with OpenSSO are:

- Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) is installed and configured.
- OpenSSO 8.0 is installed and configured
- OpenSSO Enterprise Policy Agent 3.0 for Oracle WebLogic Server/Portal 10 (weblogic_v10_agent_3) is installed and configured.
- The common prerequisite for integrating Oracle Identity Manager with third-party SSO solutions has been met, as described in "Common Prerequisites for Integration With Third-Party SSO Solutions" on page B-1.

B.2.2 Integrating Oracle Identity Manager with OpenSSO

To integrate Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) with OpenSSO 8.0 on Oracle WebLogic Server:

1. Start OpenSSO.
2. Start Oracle Identity Manager.
3. Install OpenSSO policy agent on Admin Server of Oracle Identity Manager domain. To do so:
   a. Create a J2EE agent profile on OpenSSO. Refer to the policy agent section in OpenSSO documentation for creating the profile.
   b. Install agent on WebLogic Admin Server. Install the agent by using the agentadmin utility. Refer to the policy agent section in OpenSSO documentation.
4. Install OpenSSO policy agent on Oracle Identity Manager Managed Server of Oracle Identity Manager domain. To do so, install agent on Oracle Identity Manager Managed Server. Refer to the policy agent section of OpenSSO documentation for installing the agent on a managed server. Use the same agent profile that you created in step 3.a.

   Note: For a clustered deployment of Oracle Identity Manager, install the policy agent on each Oracle Identity Manager Managed Server.

5. To configure OpenSSO policy agent after installation:

   Note: For a clustered deployment of Oracle Identity Manager, OpenSSO policy agent must be configured on each Oracle Identity Manager Managed Server.

a. Configure WebLogic Server instances with set Agent classpath and JAVA options.

b. Deploy agent application on Admin and Managed Servers.

c. Deploy and configure agent authentication provider.

d. Add WebLogic admin to bypasslist.

e. Install agent filter to oim web-apps. In this step, add OpenSSO Agent filter to all the Oracle Identity Manager web-apps that support OIM user login. To do so:

Note: The corresponding deployment-descriptors are located at:

IDM_ORACLE_HOME/server/apps/oim.ear/iam-consoles-faces.war/WEB-INF/web.xml

IDM_ORACLE_HOME/server/apps/oracle.iam.console.identity.self-service.ear/oracle.iam.console.identity.self-service.war/WEB-INF/web.xml

IDM_ORACLE_HOME/server/apps/oracle.iam.console.identity.sysadmin.ear/oracle.iam.console.identity.sysadmin.war/WEB-INF/web.xml

---

i) Go to the *IDM_ORACLE_HOME*/server/apps/ directory.

ii) Create a backup of the oim.ear/iam-consoles-faces.war/WEB-INF/web.xml file, and then edit it to add the filter element as mentioned in OpenSSO documentation. Save the changes.

iii) Create a backup of the oracle.iam.console.identity.self-service.ear file, and then extract it in a temporary location. Then extract the oracle.iam.console.identity.self-service.war file. Edit WEB-INF/web.xml to add the filter element as mentioned in OpenSSO documentation. Repackage oracle.iam.console.identity.self-service.war with the modified web.xml, and then repackage oracle.iam.console.identity.self-service.ear with modified oracle.iam.console.identity.self-service.war.

iv) Create a backup of oracle.iam.console.identity.sysadmin.ear, and then extract it in a temporary location. Then extract the oracle.iam.console.identity.sysadmin.war file. Edit WEB-INF/web.xml to add the filter element as mentioned in OpenSSO documentation. Repackage oracle.iam.console.identity.sysadmin.war with the modified web.xml, and then repackage oracle.iam.console.identity.sysadmin.ear with modified oracle.iam.console.identity.sysadmin.war.

Note: Ensure that after performing steps iii and iv, the only difference between the modified EAR files and the original EAR files is in the web.xml files.

---

v) Shutdown Oracle Identity Manager instance.

vi) Go to *OIM_DOMAIN_HOME*/servers/*OIM_SERVER_INSTANCE*/tmp/_WL_user/ directory. Go to *OIM_DOMAIN_HOME*\servers*OIM_SERVER_INSTANCE*\tmp_WL_user\ directory if the setup is on Microsoft Windows.

vii) Delete the directories specific to oracle.iam.console.identity.self-service.ear and oracle.iam.console.identity.sysadmin.ear UI applications. In a typical

Oracle Identity Manager setup, the directories to be deleted are oracle.iam.console.identity.self-service.ear_V2.0 and oracle.iam.console.identity.sysadmin.ear_V2.0.

viii) Restart Oracle Identity Manager Managed Server instance, and then check that the directories are re-created in the directory path mentioned in step vi.

6. Update the agent profile for Oracle Identity Manager Managed Server with Oracle Identity Manager URL information. To do so:

   a. Login to OpenSSO application, and select the Oracle Identity Manager Managed Server agent profile.

   b. Click the **general** tab. Change the Agent filter mode. Remove all existing values. Add new value with empty key and corresponding map value as J2EE_POLICY.

   c. Click the **applications** tab. Update the various sections as follows:

      – Login Form URI. Add the following:

```
/oim/faces/pages/Login.jspx
/identity/faces/signin
/sysadmin/faces/signin
```

      – Login Error URI. Add the following:

```
/identity/faces/signin
/sysadmin/faces/signin
/oim/faces/pages/LoginError.jspx
```

      – Not Enforced URI Processing. Add the following:

```
/identity/faces/register
/identity/faces/forgotpassword
/identity/faces/trackregistration
/identity/faces/forgotuserlogin
/identity/faces/accountlocked
/identity/adfAuthentication
/identity/afr/blank.html
/sysadmin/adfAuthentication
/sysadmin/afr/blank.html
/sysadmin/faces/noaccess
/oim/afr/blank.html
/workflowservice/*
/callbackResponseService/*
/spml-xsd/*
```

7. Configure SSO in Oracle Identity Manager. To do so:

   a. Set up WebLogic authenticators. To do so:

   i) Add and configure WebLogic authentication provider for LDAP server corresponding to the user data store used by OpenSSO. For example, if OpenSSO uses Sun DSEE, then configure iPlanet authentication provider. Set the control flag as SUFFICIENT.

Note: Ensure that all the Oracle Identity Manager users are synchronized with the LDAP server to which the authenticator points to.

---

ii) Add and configure Oracle Identity Manager signature authentication provider (OIMSignatureAuthenticator). Set the control flag as SUFFICIENT.

iii) Arrange the authenticator chain in the following order:

- DefaultAuthenticator - SUFFICIENT
- OIMSignatureAuthenticator - SUFFICIENT
- AgentAuthenticator - OPTIONAL
- LDAPAuthenticator - SUFFICIENT
- DefaultIdentityAsserter

b. Change the Oracle Identity Manager logout to execute OpenSSO logout URL by running the following command:

```
cd <IDM_ORACLE_HOME>/common/bin
./wlst.sh
connect()
addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
logouturi="http(s)://openssohost:openssoport/opensso/UI/Logout",
autologinuri="/obrar.cgi")
exit()
```

c. Set Oracle Identity Manager ssoenabled flag to true. To do so:

i) Login to Enterprise Manager. Open System Mbean Broswer.

ii) Open the oracle.iam:Location=<*OIM_SERVER_NAME*>,name=SSOConfig,type=XMLConfig.SSOConfig,XMLConfig=Config,Application=oim,ApplicationVersion=11.1.2.0.0 mbean.

iii) Set the value of ssoEnabled to true.

8. Restart Oracle Identity Manager domain.

9. Test the configuration by navigating to the following URL:

http://*OIM_HOST*:*OIM_PORT*/identity/

The page is redirected to the OpenSSO login page. Login as valid Oracle Identity Manager user.

B.2.3 Running Validation Tests to Verify the Configuration

Run the following validation steps to verify if the integration between Oracle Identity Manager and OpenSSO is successful:

User Login to Oracle Identity Manager Through SSO

Prerequisite: Create a user, for example ENDUSER001 in Oracle Identity Manager and LDAP.

Step: Try logging in to Oracle Identity Manager through SSO as the user you created, for example ENDUSER001, and check if the login is successful.

Expected output: Login is successful.

Client-Based Login to Oracle Identity Manager

Prerequisite: Make sure that Oracle Identity Manager Design Console is installed and configured.

Step: Try logging in to the Design Console as system administrator with SSO password.

Expected output: Login to the Design Console is successful, assuming that LDAPAuthenticator is configured properly for SSO login.

Signature-Based Authentication

To test signature-based authentication:

1. Try accessing the scheduler service URL. It should be running on Oracle Identity Manager Managed Server port, as shown:

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

2. Login as system administrator with SSO password.
3. If the login is successful and you can see the following details on the screen, then signature login is successful:

   Scheduler Current Status: STARTED

   Last Error: NONE

4. Click **Start** on the page if the following is displayed:

   Scheduler Current Status: STOPPED

   If no errors are displayed on the page, then signature login is successful.

B.3 Enabling Oracle Identity Manager to Work With IBM Tivoli Access Manager

This section contains the following topics:

- Prerequisites
- Integrating Oracle Identity Manager with IBM Tivoli Access Manager
- Running Validation Tests to Validate the Configuration

B.3.1 Prerequisites

The prerequisites for integrating Oracle Identity Manager with OpenSSO are:

- Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) is installed and configured.
- IBM Tivoli Access Manager (TAM) for e-business 6.1 is installed and configured.
- IBM Tivoli Access Manager Adapter for Oracle WebLogic Server for TAM 6.1 and Oracle WebLogic Server 10*g* or 11*g* are installed and configured.
- The common prerequisite for integrating Oracle Identity Manager with third-party SSO solutions has been met, as described in "Common Prerequisites for Integration With Third-Party SSO Solutions" on page B-1.
- Form based login is enabled in TAM.

B.3.2 Integrating Oracle Identity Manager with IBM Tivoli Access Manager

To integrate Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) with IBM Tivoli Access Manager for e-business 6.1:

1. Start IBM Tivoli Access Manager.

2. Start Oracle Identity Manager.
3. Setup connection between webseal and WebLogic. To do so:
   a. Create junctions to connect webseal to Oracle Identity Manager WebLogic Server.
   b. Configure webseal logout and login page.
   c. Deploy weblogic security providers.

      Refer to TAM-weblogic integration documentation provided as part of IBM Tivoli Access Manager Adapter for Oracle WebLogic Server. The additional details are as follows:

      – Keep both non-SSL and SSL ports on Oracle Identity Manager into consideration while creating junctions.
      – While creating webseal junction(s) for protected resources, make sure to use the "-c iv-user" (insert iv-user HTTP header) option.
      – List of resources that needs to be protected/unprotected:

        Protect the following resources:

        /oim

        /xlWebApp

        /Nexaweb

        /identity

        /sysadmin

        Unprotect following uris:

        /identity/faces/register

        /identity/faces/forgotpassword

        /identity/faces/trackregistration

        /identity/faces/forgotuserlogin

        /identity/faces/accountlocked

        /identity/adfAuthentication

        /identity/afr/blank.html

        /sysadmin/adfAuthentication

        /sysadmin/afr/blank.html

        /sysadmin/faces/noaccess

        /oim/afr/blank.html

        Unprotect following resources:

        /workflowservice

        /callbackResponseService

        /spml-xsd

      – Only configure Tivolli Access Manager Identity assertion provider (AMIdentityAsserterLite). Select the **iv-user** option while configuring it.
      – Do not configure Tivolli Access Manager Identity authentication provider.

- Configure WebLogic authentication provider for LDAP server corresponding to the LDAP registry used by TAM. For example, if TAM uses Sun DSEE, then configure iPlanet authentication provider. Set its control flag as SUFFICIENT. Ensure that all usersin Oracle Identity Manager are synchronized to this LDAP server. If any Oracle Identity Manager user is not present in the LDAP server, then that user will not be able to login to Oracle Identity Manager.
- Configure Oracle Identity Manager signature authentication provider (OIMSignatureAuthenticationProvider). Provide the Oracle Identity Manager database details while configuring it. You can use the same details as specified in OIMAuthenticationProvider. Set its control flag as SUFFICIENT.
- Arrange the authenticator chain in the following order:

  TAMIdentityAsserter

  OIMSignatureAuthenticator - SUFFICIENT

  LDAPAuthenticator - SUFFICIENT

  DefaultAuthenticator - SUFFICIENT

  DefaultIdentityAsserter

Note: If you cannot use TAMIdentityAsserter, then you can use the OAMIdentityAsserter, as described in "Simplifying Third-Party SSO Integration" on page B-17

4. Change the Oracle Identity Manager logout to execute TAM logout URL by using the following commands:

```
cd <IDM_ORACLE_HOME>/common/bin
./wlst.sh
connect()
addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
logouturi="http(s)://<webseal-host:port>/pkmslogout",
autologinuri="/obrar.cgi")
exit()
```

5. Set OIM ssoenabled flag to true. To do so:
   a. Login to Enterprise Manager. Open System Mbean Broswer.
   b. Open the oracle.iam:Location=<*OIM_SERVER_NAME*>,name=SSOConfig,type=XMLConfig.SSOConfig,XMLConfig=Config,Application=oim,ApplicationVersion=11.1.2.0.0 mbean.
   c. At the value of ssoEnabled to true.
6. Restart Oracle Identity Manager.
7. Test the configuration by navigating to the following URL:

   http(s)://*WEBSEAL_HOST*:*WEBSEAL_PORT*/identity/faces/home

   TAM login page is displayed. Login as valid Oracle Identity Manager user, and the login should be successful.

B.3.3 Running Validation Tests to Validate the Configuration

Run the following validation steps to verify if the integration Oracle Identity Manager and TAM is successful:

User Login to Oracle Identity Manager Through SSO

Prerequisite: Create a user, for example ENDUSER001, in Oracle Identity Manager and LDAP.

Step: Try logging in to Oracle Identity Manager through SSO as the user that you created, for example ENDUSER001, and check if the login is successful.

Expected output: Login should be successful.

Client-Based Single Login to Oracle Identity Manager

Prerequisite: Make sure that Oracle Identity Manager Design Console is installed and configured.

Step: Try logging in to the Design Console as system administrator with SSO password.

Expected output: Login to the Design console must be successful, assuming that LDAPAuthenticator is configured properly for SSO login.

Signature-Based Authentication

To test signature-based authentication:

1. Try accessing the scheduler service URL. It should be running on Oracle Identity Manager Managed Server port, as shown:

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

2. Login as system administrator by providing SSO password.
3. If the login is successful and you can see the following details on the screen, then signature login is successful:

   Scheduler Current Status: STARTED

   Last Error: NONE

4. Click **Start** on the page if the following is displayed:

   Scheduler Current Status: STOPPED

   If there are no errors on the page, then the signature login is successful.

B.4 Enabling Oracle Identity Manager to Work With CA SiteMinder

This section contains the following topics:

- Prerequisites
- Integrating Oracle Identity Manager with CA SiteMinder
- Running Validation Tests to Validate the Configuration

B.4.1 Prerequisites

The prerequisites for integrating Oracle Identity Manager with CA SiteMinder are:

- Oracle Identity Manager is installed and configured.

- CA Siteminder is installed and configured.
- The common prerequisite for integrating Oracle Identity Manager with third-party SSO solutions has been met, as described in "Common Prerequisites for Integration With Third-Party SSO Solutions" on page B-1.

B.4.2 Integrating Oracle Identity Manager with CA SiteMinder

To integrate Oracle Identity Manager with CA SiteMinder:

1. Install Siteminder WebLogic Agent by referring to Siteminder installation documentation. Follow install GUI instructions.
2. Edit the setDomainEnv.sh file to set the variables, as shown:

```
ASA_HOME='PATH_TO_SITEMINDER_AGENT_HOME'
export ASA_HOME

SMASA_CLASSPATH="$ASA_HOME/conf:$ASA_HOME/lib/smagentapi.jar:$ASA_HOME/lib/smj
avasdk2.jar:$ASA_HOME/lib/sm_jsafe.jar:$ASA_HOME/lib/smclientclasses.jar:$ASA_
HOME/lib/sm_jsafeJCE.jar"
export SMASA_CLASSPATH

SM_JAVA_OPTIONS=" -Dsmasa.home=$ASA_HOME"
export SM_JAVA_OPTIONS

CLASSPATH=${SMASA_CLASSPATH}:${CLASSPATH}
export CLASSPATH
```

3. Edit the startWebLogic.sh file to add SM_JAVA_OPTIONS to the JAVA command, as shown:

```
$JAVA_HOME/bin/java ${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME}
-Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS}
${SM_JAVA_OPTIONS} ${PROXY_SETTINGS} ${SERVER_CLASS}
```

4. Edit the *ASA_HOME*/conf/WebAgent.conf file to change the value of the EnableWebAgent parameter to YES.
5. Restart all Managed and Admin servers.
6. Add/Configure SiteminderIdentityAsserter and SiteminderAuthenticationProvider in the Weblogic authentication chain. In Identity Asserter common configuration, select SMSESSION.
7. In the Provider Specific subtab, set the "SMIdentity Asserter Config File:" field to *ASA_HOME*/conf/WebAgent.conf.
8. In SiteminderAuthenticationProvider 'ProviderSpecific', update "SMAuth Provider Config File:" to *ASA_HOME*/conf/WebAgent.conf.
9. Remove existing OIMAuthenticationProvider from the authentication chain.
10. Add OIMSignatureAuthenticator to the authentication chain. Set the control flag to SUFFICIENT. This authenticator is added only to handle signature based login to Oracle Identity Manager.
11. Add LDAP Authenticator (OID, Iplanet, and so on) to the authentication chain, and set its control flag as SUFFICIENT. Ensure that this authenticator is configured to point to the same LDAP provider, that is :
    a. Synchronized with Oracle Identity Manager, that is, have all the OIM Identity population

b. Used by the Siteminder server for authentication purposes

LDAPAuthenticator needs to be added in order to handle non-http based login requests (For example, login to OIM design console, or any other OIM client login) and OPSS based Assertion requests.

12. Rearrange the authentication chain, as listed in Table B–1:

Table B–1 Authentication Chain

Authentication Provider	Control Flag
SiteminderIdentityAsserter	
OIMSignatureAuthenticator	SUFFICIENT
SiteminderAuthenticationProvider	SUFFICIENT
LDAPAuthenticator	SUFFICIENT
DefaultAuthenticator	SUFFICIENT
DefaultIdentityAsserter	

13. Restart Admin server and all the Managed Servers in the domain.

14. Configure SSO logout for oim by using the following command:

```
cd <IDM_ORACLE_HOME>/common/bin

./wlst.sh

connect()

addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
logouturi="SITEMINDER_LOGOUT_URL", autologinuri="/obrar.cgi")

exit()
```

Note: The connect() call will ask for Admin server URL and WebLogic Admin username and password.

15. Set the ssoenabled flag for Oracle Identity Manager to true. To do so:

a. Login to Enterprise Manager, and open System MBean Browser.

b. Open the oracle.iam:Location=<*OIM_SERVER_NAME*>,name=SSOConfig,type=XMLConfig.SSOConfig,XMLConfig=Config,Application=oim,ApplicationVersion=11.1.2.0.0 mbean.

c. Set the value of ssoEnabled to true.

16. Restart Admin Server and all Managed Servers in the domain.

17. Protect/unprotect the following Oracle Identity Manager resources on Siteminder side:

- Protect following resources:

  /identity

  /sysadmin

/oim

/xlWebApp

/Nexaweb

- Unprotect the following URIs:

  /identity/faces/register

  /identity/faces/forgotpassword

  /identity/faces/trackregistration

  /identity/faces/forgotuserlogin

  /identity/faces/accountlocked

  /identity/adfAuthentication

  /identity/afr/blank.html

  /sysadmin/adfAuthentication

  /sysadmin/afr/blank.html

  /sysadmin/faces/noaccess

  /oim/afr/blank.html

- Unprotect the following resources:

  /workflowservice

  /callbackResponseService

  /spml-xsd

  /reqsvc

  /sysadmin/logout

  /identity/logout

  /identity/notification/secure

  /SchedulerService-web

  /wsm-pm

  /workflow

  /soa-infra

  /integration

  /b2b

  /sdpmessaging/userprefs-ui

18. To support client-based login to Oracle Identity Manager, the smclientclasses.jar must be added to the client classpath. To set the client classpath:

    a. Go to the *OIM_ORACLE_HOME*/server/bin/ directory using the cd command.

    b. Open the setEnv.sh file in VI Editor.

    c. Add smclientclasses.jar to the CLASSPATH variable at the end. This setting ensures successful client login to Oracle Identity Manager while executing most of the client utilities present in *OIM_ORACLE_HOME*/server/bin.

However, client classpath must be separately set for the Design Console login to work. To do so:

a. Go to the *OIM_ORACLE_HOME*/designconsole directory.

b. Open the classpath.sh file in VI Editor.

c. Add smclientclasses.jar to the CLASSPATH variable at the end.

B.4.3 Running Validation Tests to Validate the Configuration

Run the following validation steps to verify if the integration Oracle Identity Manager and CA SiteMinder is successful:

User Login to Oracle Identity Manager Through SSO

Prerequisite: Create a user, for example ENDUSER001, in Oracle Identity Manager and LDAP.

Step: Try logging in to Oracle Identity Manager through SSO as the user that you created, for example ENDUSER001, and check if the login is successful.

Expected output: Login should be successful.

Step: Try logging in to Oracle Identity Manager System Administration console (/sysadmin) as OIM Administrator (typically XELSYSADM), and check if login is successful.

Expected output: Login should be successful.

Client-Based Login to Oracle Identity Manager

Prerequisite: Make sure that Oracle Identity Manager Design Console is installed and configured.

Step: Try logging in to the Design Console as the system administrator with SSO password.

Expected output: Login to the Design console should be successful, assuming that SiteminderAuthenticationProvider is configured properly for SSO login.

Signature-Based Authentication

To test signature-based authentication:

1. Try accessing the scheduler service URL. It should be running on Oracle Identity Manager Managed Server port, as shown:

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

2. Login as system administrator by providing SSO password.

3. If the login is successful and you can see the following details on the screen, then signature login is successful:

   Scheduler Current Status: STARTED

   Last Error: NONE

4. Click **Start** on the page if the following is displayed:

   Scheduler Current Status: STOPPED

   If there are no errors on the page, then the signature login is successful.

B.5 Configuring Basic SSO Using OAM

This section describes how to configure basic integration between Oracle Identity Manager and OAM, and protect the integration with SSO authentication. It includes the following sections:

> **Note:** Performing the procedure provided in this section only enables basic SSO. Use a LDAP connector to provision passwords and also do additional configuration so that the lock status can be propagated to the directory.

- Prerequisites
- Configuring SSO Logout and the Authenticator
- Running Validation Tests to Validate the Configuration

B.5.1 Prerequisites

Perform the following prerequisites:

- Ensure that Oracle Identity Manager 11*g* Release 2 (11.1.2.3.0) is installed and configured.
- Oracle Identity Manager must be frontended with OHS/reverse-proxy, which hosts OAM 11*g* webgate.
- Ensure that Oracle Identity Manager user population is maintained in sync with LDAP repositories by using a connector. Also ensure that the Oracle Identity Manager system administrator account is created in the LDAP repository.
- Ensure that OAM 11.1.2.3.0 is installed and configured to authenticate Oracle Identity Manager users against the same LDAP repository that is synchronized with Oracle Identity Manager.

> **Note:** OIDAuthenticator is used as a reference in this procedure. If you have any other LDAP Server, such as AD, ODSEE, or OUD, then create appropriate WebLogic LDAP Authentication providers.

B.5.2 Configuring SSO Logout and the Authenticator

To configure SSO logout and the authenticator:

1. Set OIM ssoenabled flag to true. To do so:
   a. Login to Oracle Enterprise Manager, and navigate to *OIM_DOMAIN*.
   b. Right click **OIMDomain**, and select **System MBean Browser**.
   c. Click the search icon, enter `ssoconfig`, and search.
   d. In the details page, look for `SSOEnabled` flag, and select **true** from the drop down. Click **Apply** to save the configuration change.
2. Configure SSO logout for oim, as shown:

```
<IDM_ORACLE_HOME>/common/bin/wlst.sh
    connect()
    addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
logouturi="/oamsso/logout.html", autologinuri="/obrar.cgi")
```

```
exit()
```

Note: The connect() call prompts for Admin server URL and WebLogic administrator username and password.

3. Configure authentication providers. To do so:

Note: This step configures the security providers in OIM domain in such a way that the SSO login, and OIM-client based login works fine. For this, OAMIDAsserter and OIDAuthenticator must be setup. OIDAuthenticator is configured to authenticate/assert users against OID. To authenticate/assert users against any other Directory server, which is also used by OAM for authentication, corresponding authenticator must to be configured instead of OIDAuthenticator.

   a. Login to Oracle WebLogic Administrative Console, and navigate to **Security realms**, **myrealm**, **Providers**, **Authentication**.

   b. Click **New** to add `OAMIDAsserter` of type `OAMIdentityAsserter`. Click **OK**.

      Edit `OAMIDAsserter` that you added, and set the `control` flag to `REQUIRED`.

      Ensure that `Chosen Active Type` is set to `OAM_REMOTE_USER`, and then save the configuration.

   c. Click **New** to add `OIMSignatureAuthenticator` of type `OIMSignatureAuthenticator`. Click **OK**. Edit `OIMSignatureAuthenticator` and set the `Control` flag to `SUFFICIENT`. Save the configuration.

   d. Click **New** to add `OIDAuthenticator` of type `OracleInternetDirectoryAuthenticator`. Click **OK**. Edit `OIDAuthenticator` and set the `Control` flag to `SUFFICIENT`. Save the configuration. Open the Provider specific tab, and set the following attributes (only), and then save the configuration.

      – **Host:** *OID_HOST_NAME*

      – **Port:** *OID_PORT*

      – **Principal:** cn=orcladmin

      – **Credential/Confirm Credential:** orcladmin_password

      – **User Base DN:** cn=Users,dc=us,dc=oracle,dc=com

      – **All Users Filter:** (&(uid=*)(objectclass=inetOrgPerson))

      – **User From Name Filter:** (&(uid=%u)(objectclass=inetOrgPerson))

      – **UserNameAttribute:** uid

      – **User Object class:** inetOrgPerson

      – **Use retrieved use name as principal:** true

      – **Group Base DN:** cn=Groups,dc=us,dc=oracle,dc=com

      – **All groups filter:** (&(cn=*)(objectclass=groupOfUniqueNames))

      – **Group from name filter:** (&(cn=%g)(objectclass=groupOfUniqueNames))

e. Remove `OIMAuthenticationProvider` that is already configured.

f. Re-order the remaining authentication providers in the following order:

i) OAMIDAsserter

ii) OIMSignatureAuthenticator

iii) OIDAuthenticator

iv) DefaultAuthenticator

v) DefaultIdentityAsserter

g. Activate all the changes done, and then restart all the servers configured in OIM domain.

B.5.3 Running Validation Tests to Validate the Configuration

Validate the SSO logout and authenticator configuration by running the following validation tests:

User Login to Oracle Identity Manager Through SSO

Prerequisites: Create a user, for example, ENDUSER001, in Oracle Identity Manager and LDAP.

Step: Try logging in to Oracle Identity Self Service through SSO URL as the user you created, for example ENDUSER001, and check if the login is successful. Also try to login to Oracle Identity System Administration as the system administrator, and try accessing various links, such as Access Polices. Try logging out from either of the consoles, and re-login with same or different users.

Expected output: Login is successful, and all the links work as expected.

Client-Based Login to Oracle Identity Manager

Prerequisites: The Design Console is installed and configured.

Step: Try logging in to the Design Console as the system administrator with SSO password.

Expected output: Login to the Design console as the system administrator is successful, assuming that LDAPAuthenticator is configured properly for SSO login.

Signature-Based Authentication

To test signature-based authentication:

1. Try accessing the Scheduler service URL running on Oracle Identity Manager Managed server port, as shown:

   http://*OIM_HOST*:*PORT*/SchedulerService-web

2. Login as system administrator with SSO password.

3. If the login is successful and you can see the following details on the screen, then signature login is successful:

   Scheduler Current Status: `STARTED`

   Last Error: `NONE`

4. Click **Start** on the page if the following is displayed:

   Scheduler Current Status: `STOPPED`

If there are no errors on the page, then signature login is successful.

B.6 Simplifying Third-Party SSO Integration

To integrate Oracle Identity Manager with third-party SSO providers, such as Tivoli Access Manager and CA Siteminder, it is recommended to follow instructions provided in "Enabling Oracle Identity Manager to Work With IBM Tivoli Access Manager" on page B-6 and "Enabling Oracle Identity Manager to Work With CA SiteMinder" on page B-9.

WebLogic plug-ins (identity asserters or authenticators) provided by third-party SSO solutions are the recommended approach for providing SSO for Oracle Identity Manager. However, if it is not feasible to configure integration using SSO provider-specific Weblogic plug-ins, as mentioned in sections "Enabling Oracle Identity Manager to Work With IBM Tivoli Access Manager" on page B-6 and "Enabling Oracle Identity Manager to Work With CA SiteMinder" on page B-9, then instructions in this section can be followed to achieve the integration.

Note: This asserter currently supports third-party SSO providers, such as IBM Tivoli Access Manager and CA Siteminder.

To configure Oracle's Identity Asserter:

1. Login to Oracle WebLogic Administrative Console.
2. Navigate to **Security Realms**, **myrealm**, **Providers**, **Authentication**.
3. Click **New** to add `OAMIdentityAsserter`.
4. Open the asserter that you just added, and set the control flag to `REQUIRED`. In the `Active Types` shuttle, select the SSO specific HTTP header as the Chosen Active type. For example, if Siteminder SSO provider is being used, then select SM_USER header. Similarly, if Tivoli Access Manager SSO provider is being used, then select iv-user header.
5. Similarly, change the value of the SSOHeader Name field in provider-specific properties to iv-user or SM_USER appropriately.

Note:

- SM_USER and iv-user are mentioned as these seem to be the default SSO headers set by CA Siteminder and IBM Tivoli Access Manager respectively.
- For some reason, if the SSO header does not contain the username value that maps to OIM User Login field, then it is recommended to configure SSO provider to return the username as part of a header named OAM_REMOTE_USER. In this case, select OAM_REMOTE_USER as Chosen Active type in step 4, and skip step 5.

---

6. Save the configuration.
7. Configure the authentication chain as follows:

   OAMIDAsserter - REQUIRED

   OIMSignatureAuthenticator - SUFFICIENT

LDAPAuthenticator - SUFFICIENT

DefaultAuthenticator - SUFFICIENT

DefaultIdentityAsserter

Note: LDAPAuthenticator must be replaced by the appropriate authenticator that can authenticate against the LDAP provider being used by the SSO provider, for example `OIDAuthenticator`.

---

8. Configure SSO logout for Oracle Identity Manager as mentioned in sections "Enabling Oracle Identity Manager to Work With IBM Tivoli Access Manager" on page B-6 or "Enabling Oracle Identity Manager to Work With CA SiteMinder" on page B-9, based on the SSO provider.
9. Set the `ssoenabled` flag for Oracle Identity Manager to true. To do so:
   a. Login to Oracle Enterprise Manager, and open System MBean Browser.
   b. Open the oracle.iam:Location=<*OIM_SERVER_NAME*>,name=SSOConfig,type=XMLConfig.SSOConfig,XMLConfig=Config,Application=oim,ApplicationVersion=11.1.2.0.0 mbean.
   c. Set the value of `ssoEnabled` to `true`.
10. Ensure to protect/unprotect the Oracle Identity Manager resources on the SSO provider side, as mentioned in sections "Enabling Oracle Identity Manager to Work With IBM Tivoli Access Manager" on page B-6 or "Enabling Oracle Identity Manager to Work With CA SiteMinder" on page B-9, based on the SSO provider.
11. Restart all servers in the Oracle Identity Manager domain.

While using this approach of configuring Oracle's Identity Asserter, take note of the following security considerations:

- Follow standard security practices for securing OHS and WebLogic.
- Ensure that the HTTP web server front ending Oracle Identity Manager is appropriately secured by using the SSO solution's standard security practices.

B.7 Using Configurable Login ID Support for SSO Integration

Oracle Identity Manager can be integrated with third-party SSO providers, such as Siteminder and Tivoli Access Manager, in order to achieve single sign-on. These third-party SSO providers allow configuration of the login ID attribute, which the users need to use to perform SSO login. For example, if you want to allow users to login by using the email attribute (instead of User ID), then that configuration is allowed by SSO providers. However, this configuration will not work well when Oracle Identity Manager is integrated with the SSO provider. This is because the Login ID attribute in Oracle Identity Manager is `User Login`, and it is not possible to configure some other user attribute (say email) as the Login ID attribute. So, this feature is about making the Login ID attribute configurable in Oracle Identity Manager. After the login ID attribute is configured to some other user entity attribute of Oracle Identity Manager, say Email, then the users can perform SSO login to Oracle Identity Manager using the email values.

Note:

- It is not recommended to use this configuration in an Oracle Identity Manager deployment that is not integrated with SSO providers.
- This solution is recommended if your Oracle Identity Manager deployment is integrated with third-party SSO providers, and you want to allow users to login with an attribute other than User Login.
- It is not recommended to use this solution when Oracle Identity Manager is integrated with OAM. It is possible to configure OAM to allow users to login with multiple attributes, yet assert the User Login equivalent attribute. With that configuration, although the user performs SSO login using email, the JAAS subject is populated with User Login attribute.

To configure Login ID attribute in Oracle Identity Manager:

1. Login to Oracle Enterprise Manager.
2. Expand **WebLogic Domain**. Right-click ***DOMAIN_NAME***, and select **System MBean Browser**.
3. Configure the `loginMapper` property in oim configuration to use the `SSOLoginIdMapper`. To do so:
   a. Go to **Application Defined MBeans**, **oracle.iam**, **Server:*OIM_SERVER_NAME***, **Application:oim**, **XML Config**, **Config**.
   b. Change the value of the `LoginMapper` attribute to `oracle.iam.platform.auth.impl.SSOLoginIDMapper`.
4. Configure Oracle Identity Manager for SSO by setting the `ssoEnabled` attribute of `ssoConfig` to `true`. To do so:
   a. Go to **Application Defined MBeans**, **oracle.iam**, **Server:oim_server1**, **Application:oim**, **XML Config**, **XMLConfig:SSOConfig**, **SSOConfig**.
   b. Select **true** as the value of the `SSOEnabled` attribute.
5. In the same page, set the value of `loginIdAttribute` to a valid Oracle Identity Manager user entity attribute.

   Note: If loginIdAttribute is configured to Email, then all users must have a valid email ID, and the values must be unique across all the Oracle Identity Manager users.

6. For all Oracle Identity Manager users seeded by default, ensure that the value of `loginIdAttribute` is the same as that of `USR_LOGIN`. For example, if `loginIdAttribute` is configured to Email, then make sure that the email IDs of default users are the same as the `USR_LOGIN` values. The following SQL statements can be run against Oracle Identity Manager database schema:

```
update usr SET usr_email='OIMINTERNAL' where usr_login='OIMINTERNAL';
update usr SET usr_email='XELSYSADM' where usr_login='XELSYSADM';
update usr SET usr_email='WEBLOGIC' where usr_login='WEBLOGIC';
update usr SET usr_email='XELOPERATOR' where usr_login='XELOPERATOR';
```

7. Modify LDAP-specific authenticator configuration to use the appropriate attribute for User Name Attribute, User From Name Filter, and All Users Filter. For example, if loginIdAttribute is configured to Email, then make sure that the authenticator is configured as follows:

```
User Name Attribute: mail
User From Name Filter: (&(|(mail=%u)(uid=%u))(objectclass=inetOrgPerson))
All Users Filter: (&(mail=*)(objectclass=inetOrgPerson)
```

Note: User From Name Filter contains an OR condition to be able to lookup users either by using uid attribute (which is the default) or by using mail (if loginIdAttribute is configured as Email).

However, it is recommended that you perform API client-based login only by using loginIdAttribute (mail for example), if configured.

8. Create the System Administrator user entry in the LDAP provider. Ensure that the uid and mail (assuming loginIdAttribute is configured as Email) attributes are set as *SYSTEM_ADMINISTRATOR*.

Note: If the loginIdAttribute is set to some other unique attribute in Oracle Identity Manager, then the corresponding mapping attribute in LDAP must be set as *SYSTEM_ADMINISTRATOR*.

9. Perform the following changes at the OPSS layer:

Considering the fact that Oracle Identity Manager connects to SOA via HTTP (UI) as well as t3 (server) channels, you need to configure `OIMDBProvider` to handle user lookups based on the SSO Login ID, instead of the default User Login. This can be done by modifying the `idstore.oim` service instance in the jps-config.xml file as follows:

```
<serviceInstance name="idstore.oim" provider="idstore.oim.provider" location="
">
        <description>OIM Identity Store Service Instance</description>
        <property name="idstore.type" value="CUSTOM"/>
        <property name="ADF_IM_FACTORY_CLASS"
value="oracle.iam.userrole.providers.oimdb.OIMDBIdentityStoreFactory"/>
        <property name="DATASOURCE_NAME" value="jdbc/soaOIMLookupDB"/>
        <property value="USER_NAME=USR_EMAIL:USER_ID=USR_EMAIL"
name="PROPERTY_ATTRIBUTE_MAPPING"/>
</serviceInstance>
```

Note: The values for USER_NAME and USER_ID properties must be the field-mapping corresponding to `loginIdAttribute`. So if loginIdAttribute is configured as `Email`, then `USER_NAME` and `USER_ID` properties should be set to `USR_EMAIL`, since Email attribute maps to `USR_EMAIL` column.

10. Ensure that the authentication provider configuration in the Oracle Identity Manager domain security realm is as documented for that specific SSO provider, for example Enabling Oracle Identity Manager to Work With IBM Tivoli Access Manager or Enabling Oracle Identity Manager to Work With CA SiteMinder.

Note: Ensure the following while developing custom SOA composites, when a custom `loginIdAttribute` (say Email) is configured:

- When Oracle Identity Manager initiates SOA composites for approval, it passes `RequesterDetails`, `BeneficiaryDetails` as part of the payload.

  The `Login` and `ManagerLogin` fields within these would be set to Email instead of User Login.

- Ensure that you use the `loginIdAttribute` value as the task assignee.

In order to fetch the `loginIdAttribute` value for a user (given user key), you can use the `getUserDetails` operation of `RequestDataService` in the BPEL process.

The same applies to already existing custom SOA composites.

C

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 C–1 describes the grants required for database applications.

Table C–1 Role Grants for Database Applications

Role Name	Description	Oracle Identity Manager Usage	Can this Role/Grants be Removed Safely After Installation?	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.	Conditional	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	Conditional	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 C–1 (Cont.) Role Grants for Database Applications

Role Name	Description	Oracle Identity Manager Usage	Can this Role/Grants be Removed Safely After Installation?	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	Conditional	User will not be able to create any database objects. Only SYS user is 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	Yes	The user will not be able to create any views. Only SYS user is able to do so.

Table C–1 (Cont.) Role Grants for Database Applications

Role Name	Description	Oracle Identity Manager Usage	Can this Role/Grants be Removed Safely After Installation?	If Revoked
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. Oracle Identity Manager pinned sequences, function/procedures into memory. Oracle Identity Manager also pinned USR table into memory if Oracle Identity Manager has less than 50000 users in the USR table.	Conditional	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.
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	Yes	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.	No	This is required for the reconciliation engine in Oracle Identity Manager for error handling.

Table C–1 (Cont.) Role Grants for Database Applications

Role Name	Description	Oracle Identity Manager Usage	Can this Role/Grants be Removed Safely After Installation?	If Revoked
CREATE_MATERIALIZED_VIEW	Creates a materialized view in the grantee's schema	To create the OIM_RECON_CHANGES_BY_RES_MV materialized view	Yes	User will not be able to create any materialized view. Only SYS user is 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	No	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	Yes	Users will not be able to create new DDL triggers. It can be removed after schema creation.
CREATE SEQUENCE	Allows to Create sequences in the grantee's schema.	To create sequences	Conditional	Not recommended. User will not be able to create any sequence in the Oracle Identity Manager schema. Only SYS user is able to do so. **Note:** 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 which includes sequences also.

Table C–1 (Cont.) Role Grants for Database Applications

Role Name	Description	Oracle Identity Manager Usage	Can this Role/Grants be Removed Safely After Installation?	If Revoked
CREATE SYNONYM	Allows to Create synonyms in the grantee's schema.	To create the following synonyms in Oracle Identity Manager schema: ■ ALTERNATE_ADF_LOOKUP_TYPES ■ ALTERNATE_ADF_LOOKUPS ■ FND_LOOKUPS ■ FND_STANDARD_LOOKUP_TYPES	Yes	The user will not be able to create any synonym. Only SYS user is able to do so.
CTXAPP	Before you can create Oracle Text indexes and use Oracle Text PL/SQL packages, you must grant with the CTXAPP role to the grantee's schema.	To create Oracle Text indexes and Oracle Text PL/SQL in Oracle Identity Manager schema.	No	Not recommended. Oracle Text feature will not work.

Table C–1 (Cont.) Role Grants for Database Applications

Role Name	Description	Oracle Identity Manager Usage	Can this Role/Grants be Removed Safely After Installation?	If Revoked
EXECUTE ON CTXSYS.CTX_ADM EXECUTE ON CTXSYS.CTX_CLS EXECUTE ON CTXSYS.CTX_DDL EXECUTE ON CTXSYS.CTX_DOC EXECUTE ON CTXSYS.CTX_OUTPUT EXECUTE ON CTXSYS.CTX_QUERY EXECUTE ON CTXSYS.CTX_REPORT EXECUTE ON CTXSYS.CTX_THES EXECUTE ON CTXSYS.CTX_ULEXER	Oracle Text includes several packages that let you perform actions ranging from synchronizing an Oracle Text index to highlighting documents.	Oracle Identity Manager is directly consuming CTXSYS.CTX_DDL. Other may consumed indirectly.	No	Not recommended. Optimization jobs for catalog will start failing.
CREATE JOB	It grants the Create Job privileges to the grantee schema.	To create jobs in Oracle Identity Manager.	Yes	Users will not be able to create new jobs. It can be removed after schema creation.
EXECUTE ON DBMS_SCHEDULER	The DBMS_SCHEDULER package provides a collection of scheduling functions and procedures that can be called from any PL/SQL program.	To schedule the following Jobs in Oracle Identity Manager: PURGE FAST_OPTIMIZE_CAT_TAGS REBUILD_OPTIMIZE_CAT_TAGS PURGE_ADF_BC_TXN_TABLE	No	Once revoked, jobs will start failing.

Table C–1 (Cont.) Role Grants for Database Applications

Role Name	Description	Oracle Identity Manager Usage	Can this Role/Grants be Removed Safely After Installation?	If Revoked
UTL_FILE	UTL_FILE is not used by Oracle Identity Manager.	NA	NA	NA
Database scheduling using CONTROL-M	In Oracle Identity Manager, Quartz scheduler is used for application side queuing and DBMS_SCHEDULER_JOB for database jobs scheduling.	CONTROL-M is not recommended/supported by Oracle Identity Manager.	No	NA
Advance Queuing Option	Advanced QUEUE feature is used by SOA.	Used by SOA.	No	NA
CREATE ANY INDEX	Used by OPSS.	Used by OPSS.	No	NA

D

Enabling Transparent Data Encryption

This appendix describes how to configure Oracle Transparent Data Encryption (TDE) for Oracle Identity Manager.

Oracle Database supports the following types of data encryption:

- **TDE tablespace encryption:** Encrypts all content stored in that tablespace. It is useful in situations where the sensitive data are stored in multiple columns.
- **TDE column encryption:** Protects data stored in a table column. It encrypts and decrypts data transparently when data passes through the SQL layer.

> **Note:** For detailed information about TDE, see *Oracle Database Advanced Security Guide*.

Oracle Identity Manager supports and works with TDE tablespace encryption.

This appendix contains the following topics:

D.1 Configuring TDE for New Installation of Oracle Identity Manager

Configuring TDE requires downtime for the data movement from un-encrypted tablespaces to encrypted tablespaces. Therefore, you configure TDE for Oracle Identity Manager deployment immediately after installing the database schemas using Repository Creation Utility (RCU) and before installing Oracle Identity Manager application.

To configure TDE for a new installation of Oracle Identity Manager:

1. Install Oracle Database. For details, refer to Oracle Database documentation.
2. Create Oracle Identity Manager schema and the dependent schemas by running RCU. For details, refer to *Oracle Fusion Middleware Installation Guide for Identity and Access Management*.
3. Shut down Oracle Identity Manager, if applicable.

   If you are configuring TDE after installing Oracle Identity Manager, then you must shut down Oracle Identity Manager because TDE implementation does data movement and Oracle Identity Manager application will not be available for the

time period when data movement occurs from normal tablespace to TDE-enabled tablespace.

4. Create a backup of Oracle Identity Manager database schema by using the Data Pump utility.

   Using RDBMS data migration utilities, such as Data Pump, create a backup of Oracle Identity Manager database schema and the dependent schemas. This backup might be required to be restored post TDE enablement on tablespace level.

   The following is a sample command to create the backup:

```
expdp system/PASSWORD@TNS_ALIAS schemas=OIM_SCHEMA_NAME
directory=DATA_PUMP_DIR dumpfile=DUMP_FILE_NAME logfile=LOG_FILE_NAME
```

   Note: Before exporting the Oracle Identity Manager schema, capture and retain the system and object grants on it by using the following SQL commands (to be run as SYS user):

```
SELECT DBMS_METADATA.GET_GRANTED_DDL
('SYSTEM_GRANT','OIM_SCHEMA_NAME') FROM DUAL;

SELECT DBMS_METADATA.GET_GRANTED_DDL ('OBJECT_GRANT',
'OIM_SCHEMA_NAME') FROM DUAL;
```

   Copy the output of the SQL commands and edit it for appending semicolon (;) after each statement. The retained grants are required to be run post Step 10.

5. Specify the wallet location.

   You can select a directory path by specifying it in the sqlnet.ora file located in $*ORACLE_HOME*/network/admin/ directory. For instance, if you want the wallet to be in the orawallet/ directory, then include the following lines in the sqlnet.ora file:

```
ENCRYPTION_WALLET_LOCATION =
(SOURCE=
(METHOD=file)
(METHOD_DATA=
(DIRECTORY=ORACLE_HOME\orawallet)))
```

   For Oracle RAC clusters with local file system for binaries, change the SQLNET.ora of all the nodes.

   Note: A backup of the wallet location must be maintained along with the regular backups.

   To use the same Oracle database wallet share by different Oracle components, set wallet parameter as follows:

```
WALLET_LOCATION =
(SOURCE=
(METHOD=file)
(METHOD_DATA=
(DIRECTORY=D:\oracle\product\11.2.0\dbhome_1\orawallet)))
```

6. Create the wallet.

   For Oracle Database 11g:

   To use TDE, you must have the ALTER SYSTEM privilege and a valid password to the Oracle wallet. If an Oracle wallet does not exist, then a new one is created by using the password specified in the SQL command.

   To create a new master key and use TDE, run the following SQL command:

   ```
   ALTER SYSTEM SET ENCRYPTION KEY AUTHENTICATED BY "PASSWORD";
   ```

   This command performs the following:

   - Creates the wallet in the location specified in step 4.
   - Sets the password of the wallet as the one you provided. The password is case-sensitive and must be enclosed in double quotes.
   - Opens the wallet for TDE to store and retrieve the master key.

   The SQL command generates the database server master encryption key, which the server uses to encrypt the column encryption key for each table. No table column in the database can be encrypted until the master key of the server has been set.

   For Oracle Database 12c (12.1.0.2.0) Non-CDB and CDB:

   To use TDE, you must have the ADMINISTER KEY MANAGEMENT or SYSKM privilege. If an Oracle wallet does not exist, then a new one is created by using the password specified in the SQL command.

   To create a keystore:

   a. Run the following SQL command from the SYS user:

      ```
      SQL> ADMINISTER KEY MANAGEMENT CREATE KEYSTORE '<KEYSTORE_LOCATION>'
      IDENTIFIED BY <KEYSTORE_PASSWORD>;
      ```

      Here, `<KEYSTORE_LOCATION>` is the value provided in DIRECTORY path in the SQLNET.ORA file, as shown in step 5. `<KEYSTORE_PASSWORD>` is the keystore password.

      This command performs the following:

      – Creates the wallet in the location specified in step 4.
      – Sets the password of the wallet as the one you provided. The password is case-sensitive.

      The SQL command generates the database server (keystore) master key, which the server uses to encrypt the column encryption key for each table. No table column in the database can be encrypted until the keystore key of the server has been set.

   b. Verify that keystore has been created. Run the below SQL command from PDB (for CDB) and SYS (for Non-CDB) user.

      ```
      SQL> SELECT wrl_parameter,status FROM v$encryption_wallet;
      ```

      The expected result of running this SQL command:

      – Status: `Closed`
      – wrl_parameter: `<KEYSTORE_LOCATION>`

   c. Shut down the database, as shown:

```
SQL> shutdown immediate;
```

d. Restart the database by running the following command. For CDB, make sure to start the respective pluggable database(s) also.

```
SQL> startup;
```

7. Open the wallet.

For Oracle Database 11g:

As the wallet is created only once, you must specify the wallet location and create the wallet only once. The wallet must be opened explicitly with the master key whenever the database instance starts.

To load the master key after the database is restarted, run the following SQL command:

```
ALTER system SET encryption wallet OPEN authenticated BY "PASSWORD";
```

OR:

```
ALTER system SET wallet OPEN IDENTIFIED BY "PASSWORD";
```

The wallet must be open for TDE to work. If the wallet is closed, then you can access all non-encrypted columns, but not the encrypted columns.

Note: You can close the wallet by running the following command:

```
ALTER system SET encryption wallet CLOSE IDENTIFIED BY
"myPassword";
```

For Oracle Database 12c (12.1.0.2.0) Non-CDB:

As the keystore is created only once, you must specify the keystore location and create the keystore only once. The keystore must be opened explicitly with the (keystore) master key whenever the database instance starts.

To load the (keystore) master key and verify the status of the keystore:

a. To load the (keystore) master key after the database is restarted, run the following SQL command as the SYS user:

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
<KEYSTORE_PASSWORD>;
```

Here, `<KEYSTORE_PASSWORD>` is the same password used in step 6 to create the wallet.

The keystore must be open for TDE to work. If the keystore is closed, then you can access all non-encrypted columns, but not the encrypted columns.

b. Verify the status of the keystore. To do so, run the following SQL command as the SYS user:

```
SQL> SELECT status FROM v$encryption_wallet;
```

Running this command sets the status to `OPEN_NO_MASTER_KEY`.

For Oracle Database 12c (12.1.0.2.0) CDB:

As the keystore is created only once, you must specify the keystore location and create the keystore only once. The keystore must be opened explicitly with the (keystore) master key whenever the database instance starts.

To load the (keystore) master key and verify the status of the keystore:

a. To load the (keystore) master key after the database is restarted, run the following SQL command as the CDB SYS user:

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
<KEYSTORE_PASSWORD>;
```

The keystore must be open for TDE to work. If the keystore is closed, then you can access all non-encrypted columns, but not the encrypted columns.

b. Verify the status of the keystore. To do so, run the following SQL command as the CDB SYS user:

```
SQL> SELECT status FROM v$encryption_wallet;
```

Running this command sets the status to `OPEN_NO_MASTER_KEY`.

c. To load the (keystore) master key after the database is restarted, run the following SQL command as the PDB SYS user:

```
SQL> ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY
<KEYSTORE_PASSWORD>;
```

Here, `<KEYSTORE_PASSWORD>` is the same password used in step 6 to create the wallet.

The keystore must be open for TDE to work. If the keystore is closed, then you can access all non-encrypted columns, but not the encrypted columns.

d. Verify the status of the keystore. To do so, run the following SQL command as the PDB SYS user.

```
SQL> SELECT status FROM v$encryption_wallet;
```

Running this command sets the status to `OPEN_NO_MASTER_KEY`.

8. Set the master encryption key (applicable to Oracle Database 12c only).

For Oracle Database 12c (12.1.0.2.0) Non-CDB and CDB:

After the keystore is open, set the TDE master encryption key for the same. To do so:

a. To set the TDE master encryption key in a keystore, use the ADMINISTER KEY MANAGEMENT statement with the SET KEY clause. Run the following SQL command as the SYS user:

```
SQL> ADMINISTER KEY MANAGEMENT SET KEY [USING TAG '<TAG>'] IDENTIFIED BY
<PASSWORD> [WITH BACKUP [USING 'backup_identifier']];
```

Here:

- `<PASSWORD>` is the same password used in step 6 to create the wallet.
- `<TAG>` is the associated attributes and information that you define. Enclose this setting in single quotation marks (' '), for example 'oim12ccdb'.

b. Verify the status of the keystore. Run the below SQL command from SYS and PDB SYS user.

```
SQL> SELECT status FROM v$encryption_wallet;
```

Running this command sets the status to `OPEN`.

9. Drop Oracle Identity Manager and its tablespaces.

   Drop OIM user before dropping the tablespaces. The following are some sample commands to do so:

   ```
   DROP USER OIM_SCHEMA_NAME CASCADE;
   DROP TABLESPACE SCHEMA_NAME INCLUDING contents AND datafiles;
   DROP TABLESPACE SCHEMA_NAME_LOB INCLUDING contents AND datafiles;
   DROP TABLESPACE SCHEMA_NAME_ARCH_DATA INCLUDING contents AND datafiles;
   ```

10. Create TDE-enabled tablespaces and user for Oracle Identity Manager.

    Create tablespaces for Oracle Identity Manager with encryption to enable TDE at tablespace layer. You must create all the three tablespaces that you dropped in step 9. You can use DBMS_METADATA API to get the DDL for tablespace creation. The following are sample commands:

    ```
    CREATE TABLESPACE SCHEMA_NAME DATAFILE 'FILE_PATH' SIZE 128K AUTOEXTEND ON
    NEXT 64K EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO ENCRYPTION
    USING 'AES256' DEFAULT STORAGE(ENCRYPT);

    CREATE TABLESPACE SCHEMA_NAME_LOB DATAFILE 'FILE_PATH' SIZE 128K AUTOEXTEND ON
    NEXT 64K EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO ENCRYPTION
    USING 'AES256' DEFAULT STORAGE(ENCRYPT);

    CREATE TABLESPACE SCHEMA_NAME_ARCH_DATA DATAFILE 'FILE_PATH' SIZE 128K
    AUTOEXTEND ON NEXT 64K EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO
    ENCRYPTION USING 'AES256' DEFAULT STORAGE(ENCRYPT);
    ```

    Note: Datafile path can be referred from the following command:

    ```
    Select name from v$datafile;
    ```

    To validate the encryption at tablespace level, you run the following query:

    ```
    SELECT ts.name, es.encryptedts, es.encryptionalg FROM v$tablespace ts INNER
    JOIN v$encrypted_tablespaces es ON es.ts# = ts.ts#;
    ```

11. Import the data back to Oracle Identity Manager database for Oracle Identity Manager and its dependent schemas.

    As TDE enabled-tablespaces are created, you must import/restore the Oracle Identity Manager schema backup. The following is a sample command to import the Oracle Identity Manager schema backup:

    ```
    Impdp system/<PASSWORD>@TNS_ALIAS schemas=OIM_SCHEMA_NAME
    directory=DATA_PUMP_DIR dumpfile=DUMP_FILE_NAME logfile=LOG_FILE_NAME
    ```

Note:

- After importing the Oracle Identity Manager schema, execute the preserved grants, as suggested in step 4.
- After importing the Oracle Identity Manager schema, compile all the objects in the schema by using the following command (to be run as SYS user):

```
BEGIN
 UTL_RECOMP.recomp_serial('OIM_SCHEMA_NAME');
END;
```

Here, replace *OIM_SCHEMA_NAME* with the Oracle Identity Manager database schema name.

12. Configure Oracle Identity Manager.

    On successful import of the Oracle Identity Manager schema backup, continue with Oracle Identity Manager installation and configuration.

D.2 Configuring TDE for an Existing Installation of Oracle Identity Manager

If you are configuring TDE after installing Oracle Identity Manager, then perform the following steps:

1. Shut down Oracle Identity Manager because TDE implementation performs data movement and Oracle Identity Manager application will not be available for that time period.
2. Perform steps 3 through 11, as described in "Configuring TDE for New Installation of Oracle Identity Manager" on page D-1.
3. Start Oracle Identity Manager.

D.3 Deconfiguring TDE for Oracle Identity Manager

To deconfigure TDE for Oracle Identity Manager:

1. Create a backup of OIM User, tablespaces, and Object Grants by using DBMS_METADATA.GET_DDL() package.

   For information about the GET_DDL() package, refer to the following URL:

   http://www.oracle.com/technetwork/database/security/twp-transparent-data-encryption-bes-130696.pdf
2. Create a backup of Oracle Identity Manager database schema.
3. Drop OIM User.
4. Drop the following Oracle Identity Manager tablespaces:
   - DEV_OIM
   - DEV_OIM_LOB
   - DEV_OIM_ARCH_DATA
5. Close the encryption wallet by running the following query as SYSDBA user:

```
ALTER system SET encryption wallet CLOSE IDENTIFIED BY "PASSWORD";
```

6. Run the following Update query followed by Commit from SYSDBA user:

```
UPDATE TS$ SET flags=flags - 16384 WHERE online$=3 AND bitand(flags,16384) =
16384;
COMMIT;
```

7. Restart Oracle Identity Manager database.
8. Re-create OIM user, tablespaces, and Object Level grants.
9. Restore the Oracle Identity Manager backup.
10. Remove Encryption entry from SQLNET.ORA file.
11. Remove the wallet key/directory, which is *ORACLE_HOME*/orawallet/.
12. Start Oracle Identity Manager.

E

Troubleshooting Clustered OIM and Eclipselink Cache Coordination

This appendix can be used by Oracle Identity Manager administrators and developers and WebLogic administrators dealing with clustered deployments of Oracle Identity Manager and SOA. It provides some pointers to verify, test, and correct startup procedure for a clustered installation of Oracle Identity Manager and its required components if eclipselink/toplink cache coordination issues are suspected, for example, if the following exception is seen in the logs:

```
oracle.iam.platform.kernel.ProcessNotInPrePostStageException
```

It contains the following sections:

E.1 Startup Procedure for Clustered Installation of Oracle Identity Manager

For eclipselink cache co-ordination to happen successfully in a cluster environment, the clustered nodes must be started one after another. If you start all the nodes at the same time, then cache co-ordination initialization might not happen properly. In addition, the initial startup of the various managed servers is also critical, and starting servers in an incorrect order can cause some data seeding to fail.

Perform the following when starting Oracle Identity Manager, SOA, and WebLogic services:

1. Start the WebLogic Admin Server first and wait until it is in a RUNNING state.
2. Start one node of the SOA cluster and wait until it is in RUNNING state.
3. Start the next SOA Managed Server and wait until it is in RUNNING state.
4. Repeat step 3 for all SOA Managed Servers.

5. Start the first Oracle Identity Manager Managed Server and wait until it is in RUNNING state.
6. Start the next Oracle Identity Manager Managed Server and wait until it is in RUNNING state.
7. Repeat step 6 for each remaining Oracle Identity Manaager Managed Server, one at a time.
8. After all services are in RUNNING state, wait for another two minutes before permitting end user operation and other business use of the system.

E.2 Clustered Deployment Mode

Make sure `deploymentMode` is set to `cluster` in the oim-config.xml file, and MDS is updated with the changes to the oim-config.xml file. Perform an export of /db/oim-config.xml and verify to make sure it is similar to the following:

```
<deploymentConfig>
<appServerName>weblogic</appServerName>
<initialContextFactory>weblogic.jndi.WLInitialContextFactory</initialContextFactor
y>
<dataBaseType>oracle</dataBaseType>
<deploymentMode>cluster</deploymentMode>
</deploymentConfig>
```

E.3 Multicast Addressing for Oracle Identity Manager

Ensure that each cluster has its own unique multicast address configured in the oim-config.xml file within MDS. Production and test environments must not share the same multicast IP configurations. For example, the following must not share the same `multicastAddress` value:

```
<xLCacheProviderProps multicastAddress="IP_ADDRESS" size="5000">
<properties></properties>
</xLCacheProviderProps>
```

Do not share the same `multicastAddress` value as other Oracle Identity Manager deployments. If the value is used on a test domain, then do not specify the same address in the production oim-config.xml file.

E.4 Multicast Addressing for Eclipselink

Eclipselink also makes use of multicast networking for its cache coordination. Ecliplseink cache coordination uses multicast port 3121 and a Time To Live (TTL) setting of 15 hops. This is not configurable. Firewalls must take into account all multicast networking requirements of the environment.

E.5 Testing Multicast Network Testing

Refer to the following Tech notes for information about how to run some simple tests on your multicast network:

- Testing Multicast Connectivity Between OIM Nodes (Doc ID 1360763.1)
- How to Run a Multicast Monitor Test in a WebLogic Cluster (Doc ID 1064062.1)

These Tech notes can be viewed at the My Oracle Support web site at:

https://support.oracle.com

E.6 Enabling Additional Logging for Eclipselink

Add the following to the logging.xml file to enable additional logging for elcipselink/toplink:

```
<logger name='org.eclipse.persistence.session.oim.propagation' level='TRACE:32'
useParentHandlers='false'>
<handler name='odl-handler'/>
</logger>
```

Restart the domain for the changes to take effect.

E.7 Testing Multicast Connectivity Between Oracle Identity Manager Nodes

Oracle Identity Manager makes use of multicast IP network communications for normal operations. It is used by the Design Console as well as for application-level caching within Oracle Identity Manager. Sometimes, a host or router might have multicast networking disabled. This section describes how to test and verify if multicast communications between two or more nodes of an Oracle Identity Manager cluster is working.

To verify if multicast packets can be sent and received between different nodes of a WebLogic clustered Oracle Identity Manager environment:

1. Obtain the multicast IP address used by Oracle Identity Manager. To do so:
   a. Expand **Identity and Access**, **OIM**, **oim(*OIM_SERVER*)**, and from the drop-down menu on the right pane, view the System MBean Browser.
   b. View the IP Address defined in the `MulticastAddress` attribute within **Application Defined MBeans**, **oracle-iam**, *MANAGED_SERVER_NAME*, **Application: oim**, **XMLConfig**, **config**, **XMLConfig.CacheConfig.XLCacheProvider**, **XLCacheProvider**.
2. Open a command window on each host involved.
3. Go to the *MIDDLEWARE_HOME* directory.
4. Go to the coherence directory, for example coherence_3.6 or coherence_3.7, depending on which version you have.
5. If the scripts in the `bin` directory do not have execute permissions, then use the `chmod` command to enable execute permissions, as shown:

   ```
   chmod u+x bin/*.sh
   ```

6. Run the following command to start sending and receiving multicast packets:

   ```
   bin/multicast-test.sh -group MULTICAST_IP_ADDRESS:PORT
   ```

   Here, *MULTICAST_IP_ADDRESS* is the IP address obtained in step 1.

   The following is a sample output:

   ```
   Starting test on ip=iam.example.com/10.10.10.10, group=/IP_ADDRESS:12345,
   ttl=4
   Configuring multicast socket...
   Starting listener...
   Wed Feb 21 21:49:59 UTC 2015: Sent packet 1 containing 1468 bytes.
   ```

```
Wed Feb 21 21:49:59 UTC 2015: Received test packet 5 from
ip=idmsun.example.com/10.20.20.20, group=/IP_ADDRESS, ttl=4 containing 1468
bytes.
Wed Feb 21 21:50:00 UTC 2015: Received test packet 1 from self (sent 1628ms
ago).
Wed Feb 21 21:50:01 UTC 2015: Received test packet 6 from
ip=idmsun.example.com/10.20.20.20, group=/IP_ADDRESS:12345, ttl=4 containing
1468 bytes.
Wed Feb 21 21:50:02 UTC 2015: Sent packet 2 containing 1468 bytes.
Wed Feb 21 21:50:02 UTC 2015: Received test packet 2 from self
Wed Feb 21 21:50:03 UTC 2015: Received test packet 7 from
ip=idmsun.example.com/10.20.20.20, group=/IP_ADDRESS:12345, ttl=4 containing
1468 bytes.
```

Here, the multicast-test script is run on both the iam.example.com server and idmsun.example.com server using the same IP address and port number. The iam.example.com host is able to send the multicast packets and also receive the packets from idmsun.example.com host. Observing the output of the test script for a short while on each host is necessary to ensure that each host is receiving packets from all of the other hosts within the cluster. If only packets from `self` are observed, then this host is not receiving the test packets from any other systems running the multicast-test script.

Note: If the hosts are not within the number of network hops specified in Time To Live (TTL), then you can change the ttl by adding `-ttl 10` to the command.

Eclipselink also makes use of multicast networking for its cache coordination. ecliplseink cache coordination uses multicast port 3121 and a TTL setting of 15 hops. Test both the default Oracle Identity Manager multicast port as well as the eclipselink port.

Note: If a second NIC is used for multicast, then specify the interface with the `-local` attribute, such as:

multicast-test.sh -local *UNICAST_ADDRESS* -group IP_ADDRESS:12345

Where *UNICAST_ADDRESS* is the unicast address on the interface used for the multicast network. For more information see Tech note "How To Verify that Multicast Communication Works Correctly Between Machines the Coherence Cluster Members Are Running On (Doc ID 1936452.1)" on the My Oracle Support web site at:

https://support.oracle.com

If WebLogic is not used, and there is no equivalent multicast test script, then using a couple of small Java command-line applications can also be used for testing. For more information, see Tech note "How to Test Whether Multicast is Enabled on the Network (Doc ID 413783.1)" in the My Oracle Support web site at:

https://support.oracle.com